AN EMBEDDED TUTORIAL FOR A TACTICAL SYSTEM

Patrick William Flynn

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

An Embedded Tutorial For

A Tactical System

by

Patrick William Flynn

March 1980

Thesis Advisor: George A. Rahe

Approved for public release; distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Embedded Tutorial For A Tactical System		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1980
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Patrick William Flynn		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE March 1980
		13. NUMBER OF PAGES 175
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, CA 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Embedded Tutorial
Computer-aided-instruction

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Surveillance of the oceans by shore based facilities is an important segment of the ASW mission. A system with enhanced capabilities is presently being developed but problems arise in the training of Navy enlisted personnel to utilize this sophisticated equipment.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

An embedded tutorial training program preserves all the benefits of computer assisted instruction (CAI) while reducing the greatest disadvantage, cost. A demonstration of an embedded tutorial is presented which introduces the new operator to the system, provides help during operations, and has refresher exercises for experienced operators. Since a sophisticated computer-based system must provide guidance and direction during operations the introduction and refresher are available at almost no cost.

This demonstration system is intended only to show that a tutorial system can be developed and deployed and does not purport to be a complete deployable system itself.

Approved for public release; distribution unlimited

An Embedded Tutorial For A Tactical System

by

Patrick William Flynn
Lieutenant, United States Navy
B.S.M.E., Marquette University, 1974

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the
NAVAL POSTGRADUATE SCHOOL
March 1980

ABSTRACT

Surveillance of the oceans by shore based facilities is an important segment of the ASW mission. A system with enhanced capabilities is presently being developed but problems arise in the training of Navy enlisted personnel to utilize this sophisticated equipment.

An embedded tutorial training program preserves all the benefits of computer assisted instruction (CAI) while reducing the greatest disadvantage, cost. A demonstration of an embedded tutorial is presented which introduces the new operator to the system, provides help during operations, and has refresher exercises for experienced operators. Since a sophisticated computer-based system must provide guidance and direction during operations the introduction and refresher are available at almost no cost.

This demonstration system is intended only to show that a tutorial system can be developed and deployed and does not purport to be a complete deployable system itself.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

An embedded tutorial is ideal as a training device for individuals required to utilize a computer in their operational tasks because it possesses all the advantages of computer assisted instruction (CAI), while reducing its major disadvantage, cost. CAI improves the student's knowledge of the subject because it continues to give additional information until the individual achieves the desired level of performance. Studies have shown that this has resulted in higher test scores at the end of the instruction period and, in most cases, the student finishes in a shorter amount of time than that required for conventional instruction. One shortcoming of CAI has been a rise in the student attrition rate while using computer based instruction in some applications. Attrition rate studies show that the use of computer based instruction often increases the amount of student failures; however, in the case of ASW trainees the attrition rate is decreased. This could be traced to the fact that the selectees for ASW training are of a slightly higher initial aptitude and are more inclined towards use of a computer since their operational jobs involve the use of some type of computer. Apparently, a machinist mate being trained on the repair of a pump would feel frustrated in not being able to get his hands on the equipment and thus would reject the use of this type of training.

The embedded tutorial described in this paper demonstrates the capabilities of a computer and a color graphics console in the training of a prospective ASW tracker. It is only a demonstration because the actual system is not fully developed, the actual type of color graphics console was not available, and the actual data for test cases were unattainable.

The numbers throughout this paper fall into three general classifications, common knowledge values, general range values, and specific constraints. The common knowledge values are those numbers that can be found in any basic textbook, such as the speed of sound in water, and thus are not classified. The general range values are those where a spectrum of values could be sighted and thus a random value was picked, for example 41.8 as the frequency of the signal being detected. The last numbers used throughout this paper are a series of XXX's that are substituted for values which could not be revealed for security reasons. The actual values would be included in the operational tutorial.

The tutorial permits the user to proceed in any of three areas of study depending on his abilities and previous training. There is a basic tutorial on the overall system, the environmental problems, and the formulas used to tranform the data into useful displays. Next, there is a short tutorial on the types of data displays that a tracker will use in the operational

system. Last, there is a set of scenarios that can be used to refresh the skills necessary to effectively track vessels.

II. EMBEDDED TUTORIALS

A. PURPOSE OF EMBEDDED TUTORIALS

An embedded tutorial, as discussed in this paper, means the use of computer aided instruction (CAI) on the computer and graphics terminals that are actually used to track targets. The purpose of embedding the tutorial in the operational system is three-fold: first, the individual being trained becomes accustomed to the surroundings and the response time of the terminal; second, since the computer and terminals are already present and there is room in memory to store the tutorial, this procedure is more cost-efficient than a separate system or conventional training methods; third, it provides a reference manual for an operator actually using the system. There are many more subtle benefits in using an embedded tutorial. By training an individual, at least partially, on the actual operational system a great deal of the anxiety of a young, inexperienced operator can be relieved. The operator will become familiar with the computer, the system, and their idiosyncracies, and thus be more comfortable when the pressure of the mission begins.

An embedded tutorial maintains all the advantages of computer based instruction while reducing one of its major disadvantages, cost. Since the tutorial is part of the

operational system, a major part of the cost is eliminated. Even if some additional memory and terminals have to be installed, great savings are achieved. The data collected by several surveys shows that even with stand-alone computer-based instructional systems the money saved because of reduced training time balances the cost of the system.

While manuals could be provided for the user to review when a problem arises on the operational system, it would be more beneficial for the user to see the solution or course of action on the terminal being used at that time. A picture of a surface plot in a book cannot compare to one on a color graphics terminal and the computer can produce a vast amount of variations corresponding to the particular problem. Since the embedded tutorial is attached to the operational data, the reference material could be kept constantly up to date.

B. SYSTEM SETUP

This tutorial was designed to demonstrate some of the capabilities of a color graphics system to provide a computer-based instructional package. While the Ramtek 100A used for this study performs many of the desired functions: easy color manipulation, simple input devices, quick response to user input, and fair picture resolution, the proposed Ramtek 9400 system has superior characteristics. It out performs the 100A system in speed, resolution, and other capabilities. The Ramtek 9400 is on

the order of ten times faster than the present system. It also has 32K x 32K virtual picture and a 1024 x 1280 screen resolution which enables the displays to appear much more lifelike. Some of the capabilities of the Ramtek 9400 that enable the designer of a tutorial more flexibility are: panning, zooming, blinking, rotation in 90-degree increments, scrolling up, down, right, and left within arbitrary rectangular limits, interrupts, and delays. Since all of these function are incorporated in the machine, the coding of a tutorial or any program will be greatly reduced. A Ramtek 9400 is presently being installed at the Naval Postgraduate School Computer Laboratory, which will allow the further development of this tutorial system.

C. ADVANTAGES / DISADVANTAGES

1. Student Achievement

The effectivenss of computer aided instruction compared to conventional instruction has been measured only by student performance and not by on-the-job performance after graduation. Student achievement might be an aid in predicting quality of performance on the job but the correlation between these two measures has not been established for any of the training methods. Student achievements have shown that computer aided instruction is as good or better than other methods in practically all cases. The fact that student achievement with computer aided instruction (CAI) is better than conventional

instruction or individualized instruction may also be a direct consequence of the fact that students instructed by CAI remain in these courses until they have mastered all the lessons. The critical variable thus becomes the amount of time needed to complete courses given by computer-based instruction. An additional benefit is achieved by ensuring that an individual has mastered all of the lessons needed to be a qualified tracker in the fleet.

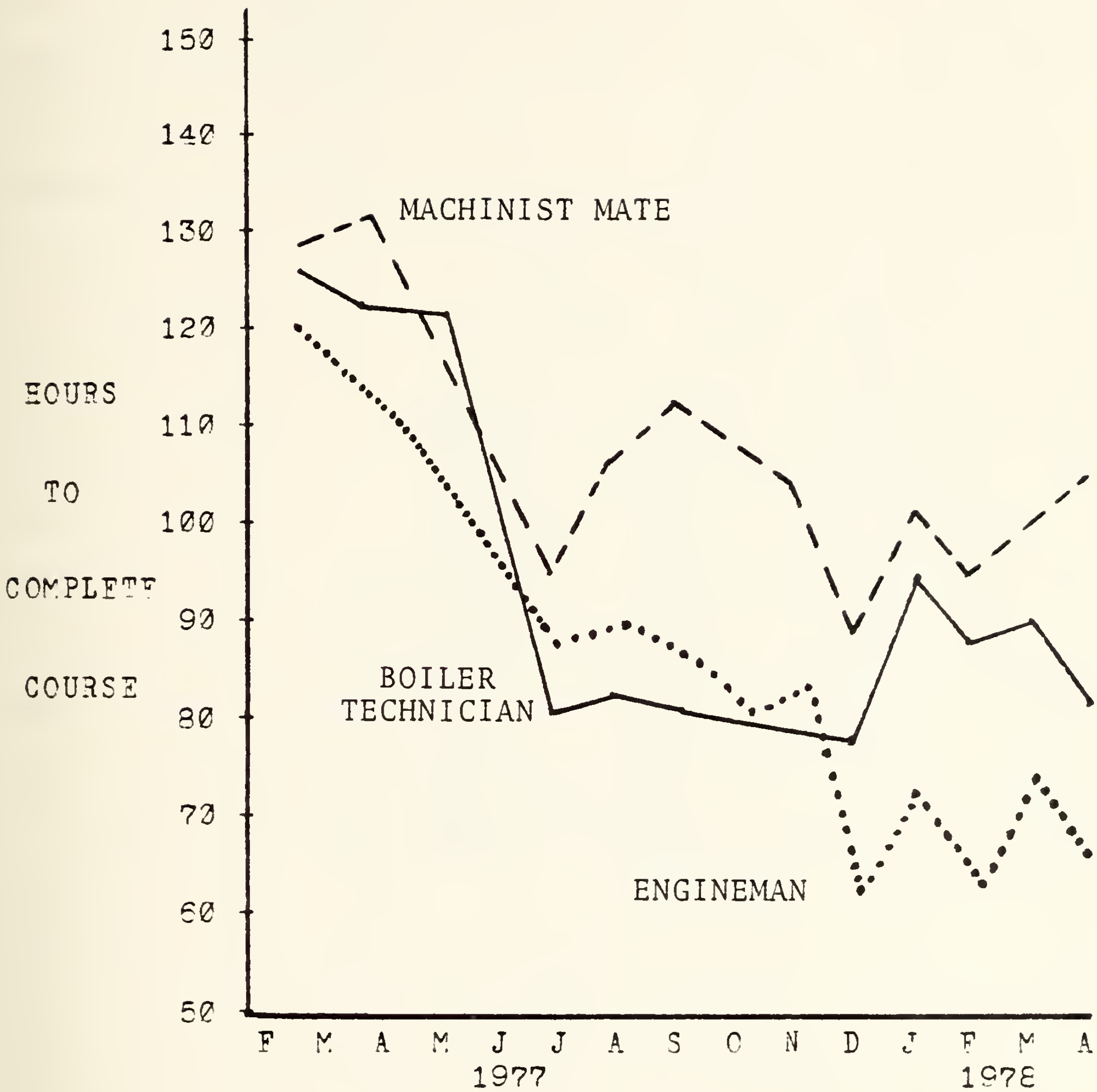
2. Student Time Savings

Findings in a report, IDA paper P-1375, prepared for the Office of the Under Secretary of Defense for Research and Engineering shows that computer-based instruction saves about one-third of the time required for courses using conventional instruction; however, there are wide variations in the amounts of savings that have been reported. Two major uncontrolled variables in these studies are the unknown quality of the instructional materials used in the various comparisons and uncertainty that the same amounts of course materials were used in both methods of instruction.

The fact that CAI saves time is consistent with well-known information about the effect of great differences both in student abilities and in knowledge of the subject at the start of any course. In conventional instruction with a fixed amount of time, these differences lead to variations in the amounts of knowledge acquired by the end of the course, as shown by a distribution of final

grades. In individualized instruction, whether computer-based or not, each student proceeds at his own pace and differences between students influence the amount of time needed to complete the course more than it does the amount of information acquired. Most of the time savings in individualized instruction are due to those students whose rate of progress in conventional instruction would be too slow; typically, that rate might be one that permits about 90 percent of the students to complete the course during the fixed period of time.

An example of the time savings can be seen in the graph (figure 2-1) which shows the reductions in student time for about 12,000 graduates over a 15-month period ending May 1978. The results were for three courses on the Navy Computer Managed Instruction System at Naval Air Technical Training Center, Millington, Tennessee.

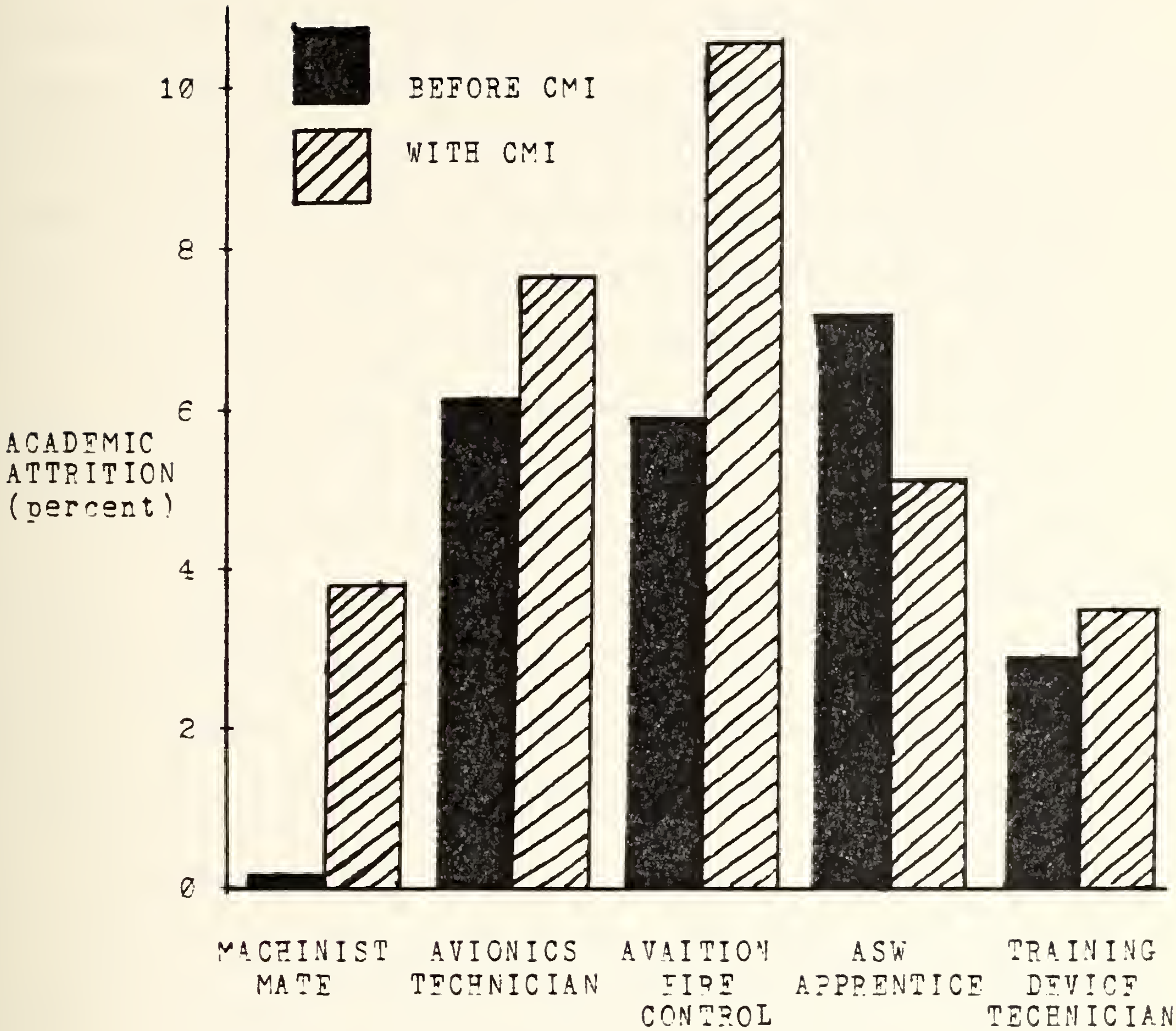


STUDENT TIME-SAVINGS

FIGURE 2-1

3. Student Attrition

Since the method of instruction may influence the number of students who can sucessfully complete a course, the rate of academic attrition associated with alternative methods of instruction is a matter of concern. The rate of attrition is a measure of the cost of instruction since it influences the number of students needed to enter a course in order to produce a specified number of graduates. Only two computer-based instructional systems, the Air Force AIS and Navy CMI, have received extended, though still limited, use in military training. Academic attrition may have increased in courses taught this way, compared to attrition with conventional instruction during prior periods, but the abilities of the graduates were significantly higher. Also since these comparisons do not take into account possible changes in the qualifications of students over the same time periods, the available data suggests, but does not prove, that computer-based instruction may increase academic attrition over that found with conventional instruction. Figure 2-2 shows the attrition rates of seven courses taught using both methods of instruction.



STUDENT ATTRITION RATES

FIGURE 2-2

4. Attitudes of Students and Instructors

Students almost always favor computer-based instruction over the conventional instruction, while instructors almost always have unfavorable attitudes towards the CAI instruction. The students attidudes are generally due to two factors: excitement about the use of a new type of machine and better control over the material that is being taught. Both of the factors are constantly at work and thus by the time the novelty of the system wears off the student generally has developed a strong attitude that he has control over his academic destiny. The instructors, on the other hand, dislike the new system because their professional training is still largely oriented towards conventional instruction and those that are assigned to computer-based instructional duties receive little guidance on how to conduct such courses.

D. COST-EFFECTIVENESS

There is no evidence that one method of instruction is most cost-effective for all types of military training. The most cost-effective method for a particular situation will depend upon such factors as type of course material, location of instruction, number of students, and life-span of the training. It is apparent that the lack of cost data currently available for this system makes it impossible to examine satisfactorily the conditions which would make a particular method of instruction the most cost-effective alternative since the system is still in the research

stages. However, the type of course material seems ideal for a computer-based instruction since the trainee will be using the computer in the execution of his duties. The location of the instruction in an embedded tutorial is ideal because it is present in his work space so that he is able to refresh his skills at any time. While the number of students utilizing the instruction has a bearing on the cost-effectiveness of the instruction, the embedded tutorial reduces or eliminates the cost of the computer. Since this system already has the computer for operational purposes, the tutorial can be stored and operated on that computer, with the possibility of purchasing only one additional console. The life-span for this system looks favorable for computer-based instruction since it is a new system that should be around for an extended time. Since it is a new system, the instructions will constantly have to be updated which is ideal for a computer-based system as compared to a textbook that would have to be reprinted.

Other benefits, beyond those mentioned above, occur with computer-based instruction largely because the computer can compile records and direct the attention of instructors, on the basis of various algorithms. The following list of benefits is illustrative rather than complete:

1. More precise data for improving and updating course materials;

2. Improved control over equipment, facilities, and

materials for instruction;

3. Improved allocation of resources among students;

4. Improved ability to accommodate fluctuations in student loads;

5. Increased student/instructor ratios, as well as the ability to use some instructors with less advanced qualifications;

6. Reduced need for support by noninstructional personnel;

7. Reduced time of students on base waiting for courses to start;

8. Improved integration of records of students at school with those in central, computer-based personnel files; and

9. Improved utilization of instructors.

III. TUTORIAL

The embedded tutorial, as described in the following chapters, is designed to train an inexperienced individual, provide basic data information, and enhance the skills of an experienced operator. The three separate sections of the tutorial provide the versatility necessary to make this system useful because it covers the entire spectrum of training. As a new recruit is assigned to the station it would be advantageous to indoctrinate him utilizing the actual operational system. After a thorough review of the basic system, the user can start to get a feeling for the type of data to be processed by continuing the tutorial in the data display section. When the user is ready to actually use the operational system, it would be advantageous to first practice on some old data which can be done by utilizing the scenario section of the tutorial.

The displays presented throughout this paper (figures 3-1 to 6-17) are representations of the actual displays that are seen on the operational tutorial. The figures do not adequately reflect the actual display because the operational system is on a color graphics console. The outer border of each figure represents the limits of the screen, while each horizontal line depicts a change of color. For example, figure 3-2 is broken into three sections by the two horizontal lines.

A. INTRODUCTION DISPLAY

The Embedded Tutorial Introduction display (figure 3-1) provides the user with the option areas available. The user, dependent on his skills, can choose to follow any of the three sections of instruction or quit the entire tutorial. If he chooses to review the basic principles of the system (option a), then he will receive the fundamentals of the system, some of the environmental problems, and a little theory on correlation techniques. This section of the tutorial flows through the basic components of the system but allows an inquisitive individual the ability to delve into some additional details. The tutorial starts off with a simple introduction followed by the overall view of what the system is trying to accomplish. Next a list of problem areas related to the environment of system operation is provided with the option to further examine each of these problem areas. When the user has completed the environmental problems, a description of the correlation technique is discussed followed by a brief exhibition of the complex machinery of the system. Upon completion of this section of the tutorial the program will return to the display (figure 3-1) in the main program and allow the user to again proceed along any of the three routes, basic principles of the system, data information displays, or scenarios.

EMBEDDED TUTORIAL INTRODUCTION

This tutorial is designed to meet the needs of an assortment of users, from the novice to the experienced operator. There are three categories that can be reviewed:

a) The basic principles of the system are discussed and many of the problems associated with the system's environment are dealt with in this section.

b) The type of data that will be displayed during actual system operation is presented along with explanations of the data and key areas to watch while tracking.

c) A set of practice scenarios are presented based on the parameters inserted by the operator. These scenarios allow the user to see actual data being presented and the effects of various parameter settings. Key points are also highlighted in this section.

type in the letter of the desired category or (q) to quit.

INTRODUCTION DISPLAY

FIGURE 3-1

If the individual decides that the data information displays (option b) is the area that needs to be covered next, then an assortment of data in the format that will be used in the operational case will be presented. This data includes: estimated positions, courses, speeds, probabilities of accuracy, tau values, doppler values, and surface plots of the suspected track. These are only example displays to illustrate the format of the data and to explain the meaning of various values. Key areas of interest are also explained in the lower section of each display.

When the user is ready to tackle the operational tasks, data that has already been processed is provided in the scenario section (option c) to enhance capabilities while keeping the operational data safe from accidental removal. The scenario section allows the user to insert the parameters and then displays the actual data in the same format as the operational system. It also gives a very brief explanation of the reason for the outcome of the data; i.e. The data is variant due to the low coherence level. With this information the user can return to the parameter setting display and adjust the values to obtain a better set of data. This is very useful in a new system because there are no values to which the parameters can be set which automatically achieve the best results each time. The scenario section is a good starting place and also an excellent place for an experienced operator to

improve performance and practice different types of parameter settings without damaging the operational data or downgrading operational performance.

B. INSTRUCTIONAL DISPLAY

The instructional displays, an example is figure 3-2, provide the user with the basic understanding of how each display will be set up. Each section has slightly different characteristics so there is a different instructional display to explain each section of the tutorial.

Each instructional display follows a basic pattern: menu, illustration, and explanation. The menu provides the user with the possible options available to him at any time. They include: "Continue" which will proceed to the next basic illustration; "Return" which will return the program to the initial display; and, option categories to proceed to any of the possible areas of further detail desired at that time. The illustration area of the display varies from a picture of the situation, to a display of data, to the correlation formulas. The illustrations try to cover as much information as can normally be digested in a single frame and then the explanation section tries to clarify any additional problems. The explanation occasionally causes the user to be uncertain about some point. Additional information can generally be obtained by proceeding to one of the option areas.

Menu of the next possible option areas is presented here
Basic diagrams will be illustrated in this area. They will depict a view of the overall system, environmental problems, and detailed displays of questionable areas. Review of this section before reading the explanation below will be helpful in succeeding displays.
An explanation of the middle area will be displayed in this section. Generally it will be a simplified review of the basic concept with a few new ideas. If the new ideas are not fully understood, they will be listed in the menu to be pursued later. type c to continue

INSTRUCTION DISPLAY

FIGURE 3-2

IV. BASIC SYSTEM TUTORIAL

This chapter describes an embedded tutorial written for an ASW system presently being developed for the United States Navy. This section of the tutorial is designed to indoctrinate a new operator in the basic characteristics and problems of the system.

A. INTRODUCTION DISPLAY

The introduction display (figure 4-1) provides the user with the basic concepts necessary to understand the system's purpose. It prepares the user for the general flow of the tutorial and the option areas that will be available for further study. If an individual only types "c" to proceed to the next display, then a basic overview of the system and its environmental problems will be presented. However, the user has the option to branch into the greater details of many of the problem areas and formulas.

B. OVERALL SYSTEM

The second display (figure 4-2) familiarizes the user with the purpose of the system, provides a possible configuration of arrays, and the target area that is to be covered. There is no greater detail available at this point, so the only options are to continue or return. If the operator chooses to return, a new display will be presented which asks him 'DO YOU REALLY WANT TO RETURN TO THE MAIN

PROGRAM? (y / n)'. If he responds 'y' the program will shift back to the main program, but if anything else is typed the program will return the previous display to continue.

C. ENVIRONMENTAL PROBLEMS

When the continue option is taken a new display (figure 4-3) appears on the screen describing the problems associated with the environment that the system must work in to detect targets. It discusses the problem of extraneous noise always present in the ocean, the time delays in receiving the signals due to long distances, and the problem of sound traveling at various speeds. All of these problems are touched on in this one display, with further explanation in subsequent displays.

The next set of displays demonstrates one area where a computer based embedded tutorial is far more proficient in training than a normal textbook could possibly be. There are six displays, figure 4-4 through figure 4-9, which can be presented from this point and then repeatedly cycled through in any order desired by the user. Presently there are only six displays, but this is only a sample of an embedded tutorial, there could be 20 or 30 or more displays open to the user to traverse in any manner desired. This allows the user to control the order of areas of learning corresponding to his interests and curiousity.

The first environmental problem that the system must overcome is the presence of vast amounts of noise always present in the ocean. In the frequency ranges that the

surveillance system is concerned with, the principle source of noise is other ships in the ocean. Once the noise is generated it will travel great distances but will dissipate and thus be useless in detection. It is very similar to trying to listen to a single person in a large noisy party; the person's voice is distinguishable but the background noise makes it rather difficult to clearly understand it. Figure 4-4 illustrates how the desired frequency of the vessel stands out with the noise slightly lower.

If 'd' was typed while in any of the displays (figures 4-3 thru 4-9) a presentation of the problem associated with large distances would appear on the screen. This display (figure 4-5) compares two possible arrays trying to detect a target in a specific area, where one is 1200 miles and the other is 500 miles from the source of the signal. This creates two problems. One is due to the large difference in distances which causes time delay problems (figure 4-8), and the other problem is due to tau line separation (figure 4-6 and 4-7). At this point the user can choose to follow up on either problem or possibly one of the other four options.

The tau lines display (figure 4-6) describes what the tau lines are and how they are involved in the system. Tau lines are a time delay between reception of the same signal by two different arrays. Thus if a signal is produced anywhere on a line equidistant from both arrays, it will reach the arrays at the same time, this line is the zero tau line. Likewise, any signal produced on line 3-4 (figure 4-6)

will arrive at array "A" 500 seconds prior to the same signal arriving at array "B". As line 3-4 demonstrates, these tau lines are hyperbolic in shape and they get more bent as the line moves closer to either array. This bending results in the distance between any two tau lines not being constant. As the tau line problem display (figure 4-7) illustrates, separation of lines near point 4 is nearly 225 miles, while they began only 50 miles apart. Another problem is that, over the same distance, different tau lines, i.e. 1-2, only separate to approximately 65 miles. Thus, depending on the position of the area to be covered, accuracy can diminish quite drastically. From this point in the tutorial, the user can examine more problems, review this problem, or continue on to the next general area.

If the user decides to proceed with some new problem areas, two more have been implemented, time delay and water temperature. The time delay display (figure 4-8) again shows the basic picture of the surveillance area, but then discusses a new problem. It explains that a signal produced at point "A" can arrive at two different arrays separated by a large time difference. This problem has to be accounted for by the computer.

The last display in this section of problems deals with the speed of sound in water and how temperature, salinity, and depth effect the speed. The display (figure 4-9) illustrates the variance of speed with depth. As the depth is increased the speed of sound initially slows down until

roughly the 1000 yard depth and then it starts to increase gradually. As explained in the write-up under the graph, speed of sound also increases as the temperature and salinity increase. Again the user has the opportunity to continue reviewing any of these problem areas, continuing the tutorial, or returning to the main program.

D. SYSTEM BREAKDOWN

Continuing the tutorial presents the user with an estimate of the area this system covers. The display (figure 4-10) illustrates that the coverage is about the size of the state of Montana, for an arbitrary operating distance in this example in excess of 1500 miles. This display also shows that the area can be broken down into smaller, more manageable regions of about 40 miles square. Thus there are 100 regions in this surveillance area and they can be covered by XXX correlations per hour. The XXX was purposely displayed to show where the vast abilities of the computer could be shown.

BASIC TUTORIAL INTRODUCTION

This embedded tutorial is designed to provide the novice user with an introduction to the ASW surveillance system. The system is designed to detect vessels at extreme distances, in excess of 1000 miles, in order to protect our coast and shipping lanes from enemy submarines and other ships.

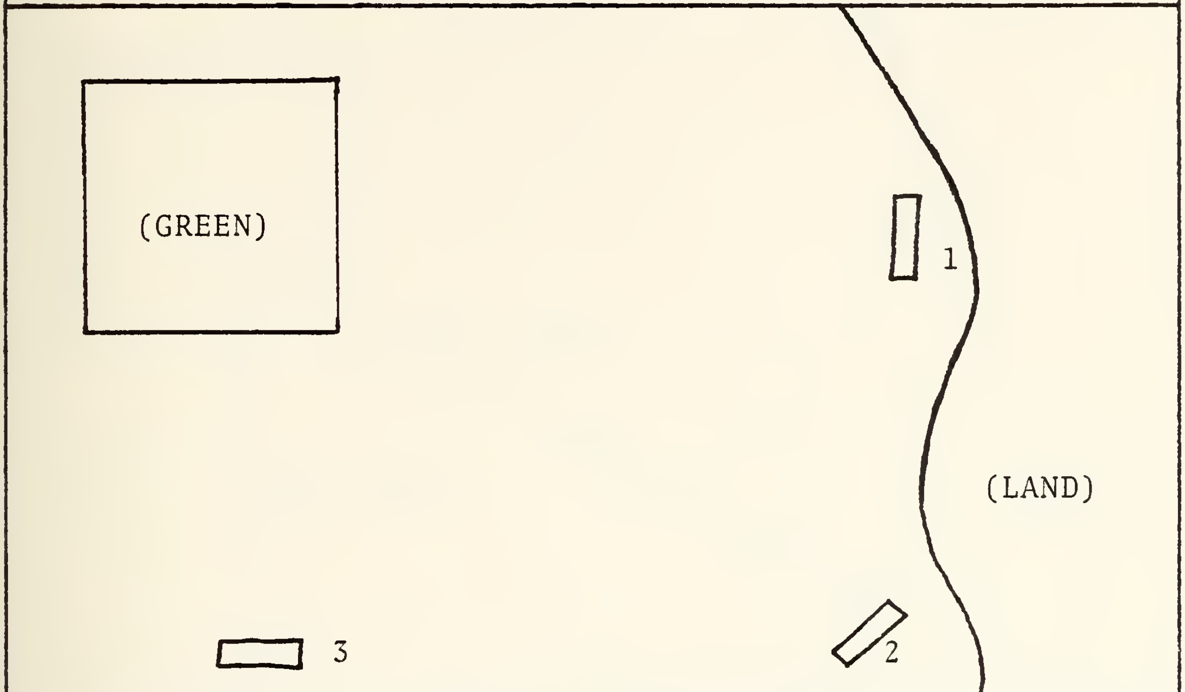
Many of the following displays exhibit the fundamental concepts of the system and then allow the user to delve into the details when desired. Observe the menu at the top of each display to see what options are available at any given time.

type c to continue

FIGURE 4-1

BASIC TUTORIAL INTRODUCTION

type ==> Continue, Return

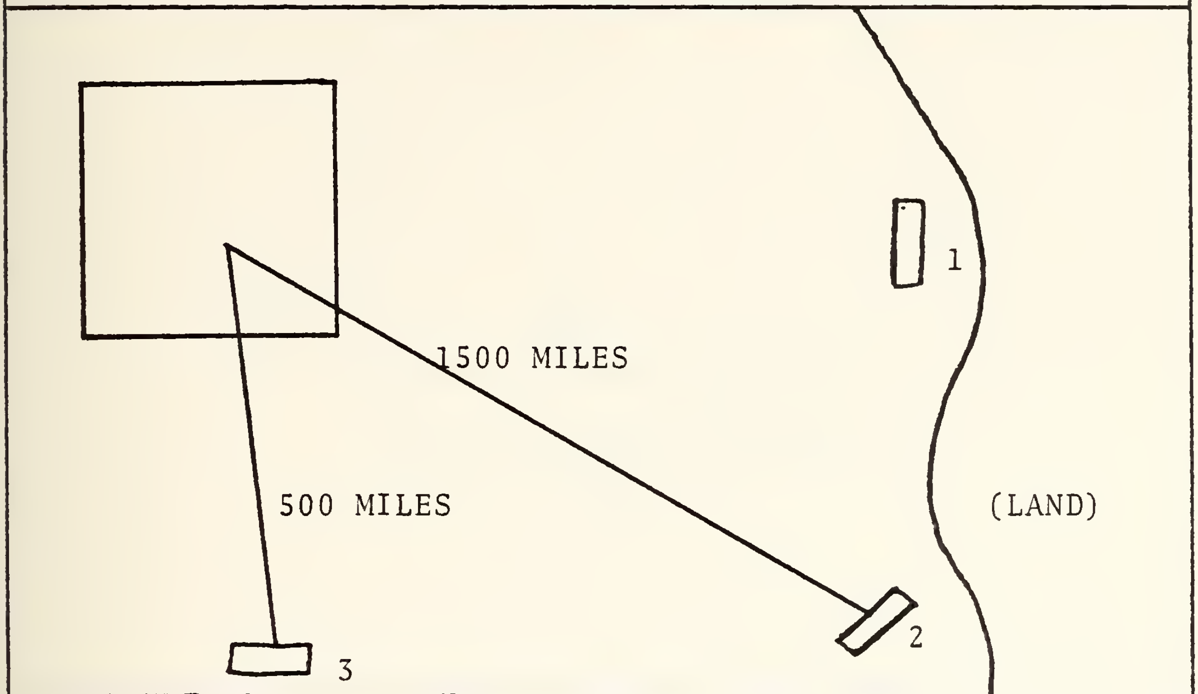


The surveillance system is designed to detect signals produced by targets and then determine the position of the targets. Assuming there was an area (green square) in the ocean to be watched for submarines, the two arrays of hydrophones (1 and 2) and a method to analyze their outputs would be necessary. For better results additional arrays such as 3 could be added.

FIGURE 4-2

OVERALL SYSTEM VIEW

type==> Continue, Distance, Time, Water(temp), Noise, Return



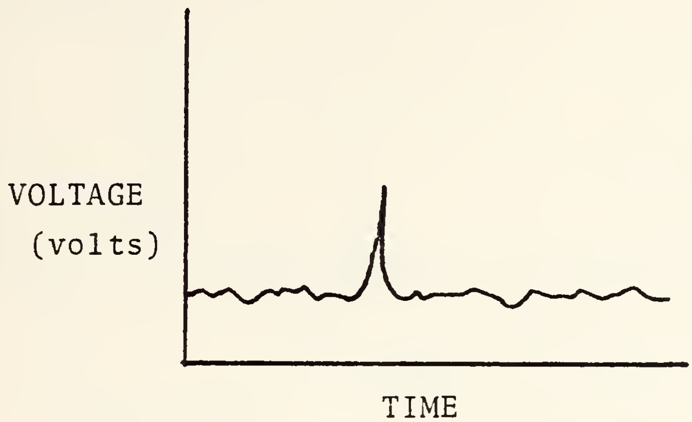
The above menu lists many of the problems associated with the system due to its environment:

1) Distances are quite large, on the order of 1500 miles in this example.
2) Time delays caused by differences in the time of arrival of the signal at two different arrays.
3) Water temperature and salinity variations affect the speed of sound.
4) Noise created by other vessels distorts the reception of the desired signal.

FIGURE 4-3

ENVIRONMENTAL PROBLEMS

type ==> Continue, Distance, Time, Water(temp), Return

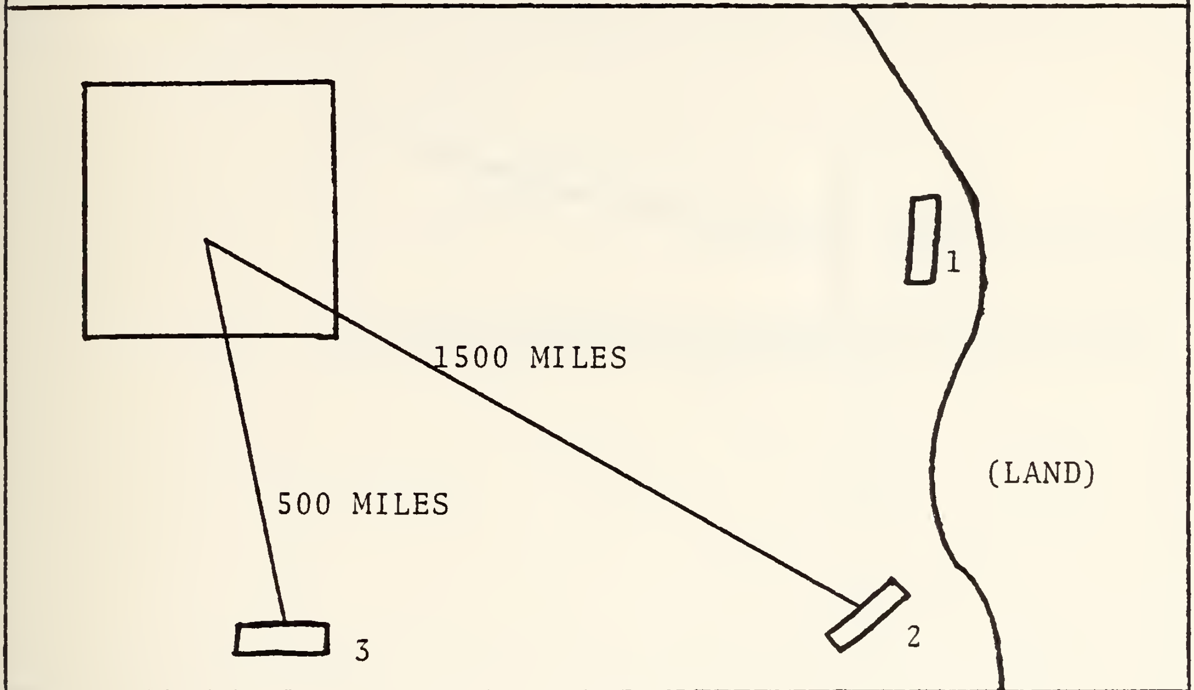


The detection of a specific signal is adversely affected by additional noise. This detection can be compared to trying to listen to a specific person at a large noisy party. The majority of signals detected in the range of 10Hz to 400Hz are generated by other vessels. However, greater distances cause these signals to be weaker and untraceable.

FIGURE 4-4

NOISE PROBLEMS

type=>Continue, Time, Water(temp), Lines(tau), Noise, Return

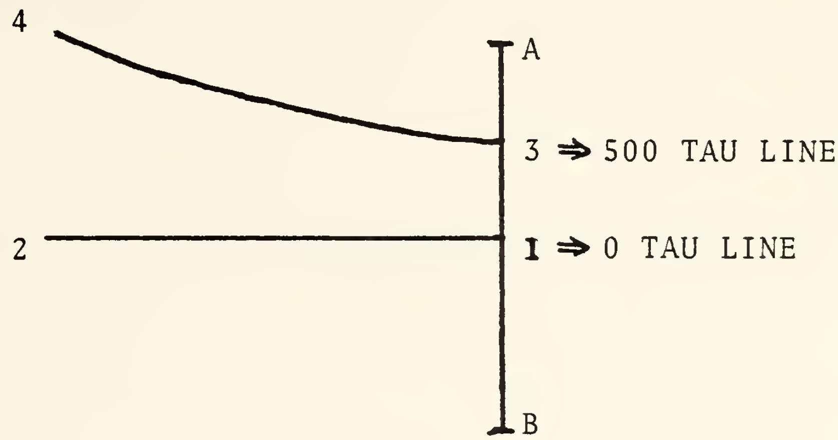


This system is required to detect targets at very large distances. These distances range from relatively few miles to in excess of 1500 miles in this example. The long ranges cause problems of diffusion and large separations in the tau lines. When the detection of a signal is accomplished by two arrays that are at different distances from the source of the signal, a time delay must be computed in order to pinpoint the position of the source.

FIGURE 4-5

DISTANCE PROBLEM

type=>Continue, Noise, Time, Water(temp), Line(prob), Return

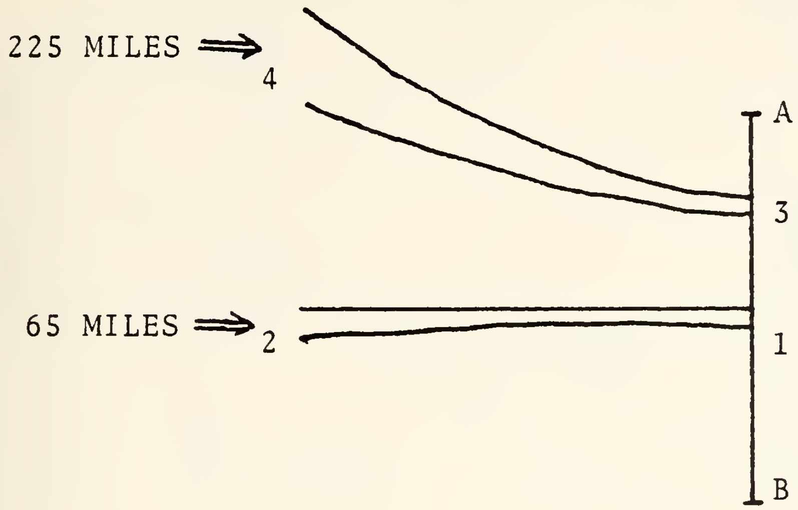


Tau lines are measurement lines distinguishing the time arrival of the signal at two different arrays. If arrays A and B are 1000 miles apart then a signal produced midway between A and B (point 1) would arrive at both arrays at exactly the same time. Additionally any signal produced on line 1-2 would arrive at both arrays simultaneously. In the same manner, signals produced on line 3-4 arrive at the arrays with the same amount of time delay in all cases.

FIGURE 4-6

TAU LINES

type==> Continue, Noise, Time, Distance, Water(temp), Return

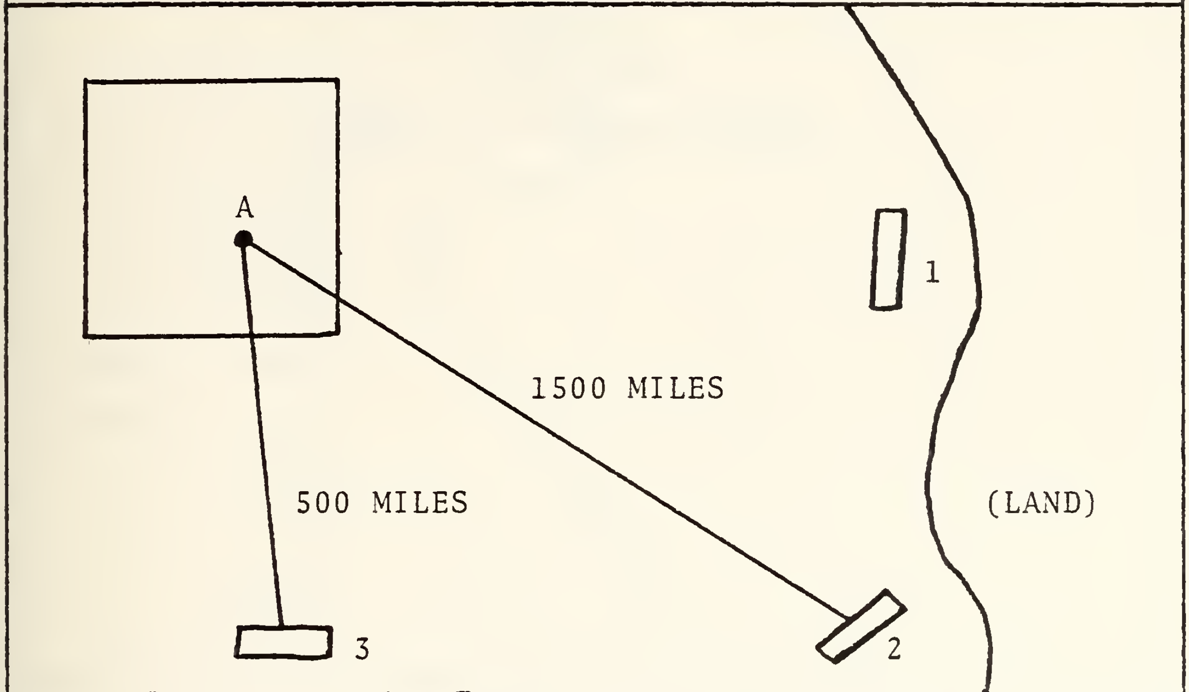


Since the tau lines are hyperbolic in shape, another problem arises in that the spacing between any two lines does not remain constant. The lines at points 1 and 3 are spaced 50 miles apart. The lines 1-2 separate to 65 miles by the time they reach point 2, whereas lines 3-4 spread to 225 miles by the time they reach point 4. This variance can cause drastic differences in the accuracy of the signal received, depending on the position relative to the arrays.

FIGURE 4-7

TAU LINE PROBLEMS

type ==> Continue, Noise, Distance, Water(temp), Return

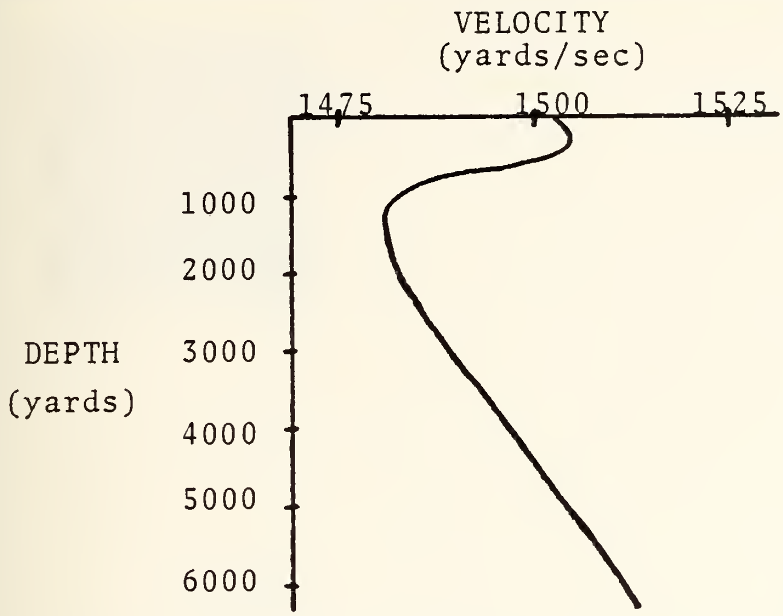


Since sound travels at 2989 Kts. (3400 miles/hour), a signal could be considered to travel at one mile per second. At this speed it would take a signal 1500 sec. or 25 minutes to travel from point A to array 2. But it would only take 8 minutes for it to travel to array 3, thus arriving 17 minutes before it would arrive at array 2. This difference must be accounted for by the algorithms in the computer.

FIGURE 4-8

TIME-DELAY PROBLEMS

type ==> Continue, Noise, Distance, Time, Return

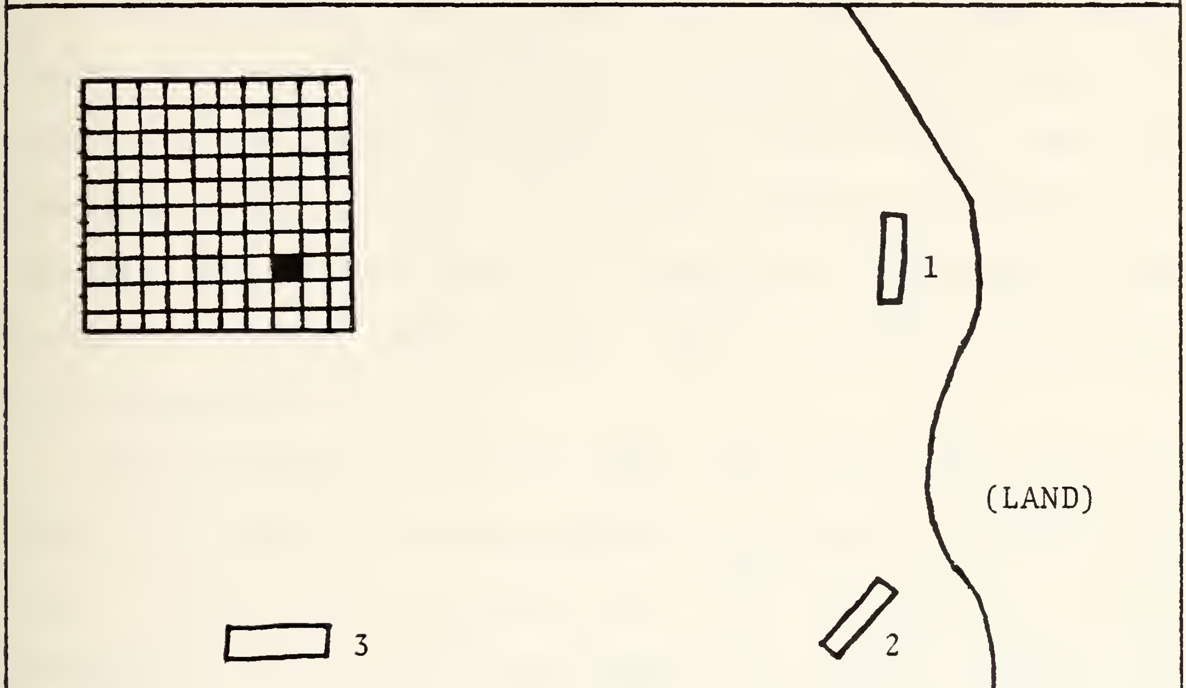


The velocity of sound varies as a function of the water temperature, salinity, and depth. In the first 300 yards from the surface, velocity is mainly affected by temperature and salinity, after which depth (pressure) is the main factor. As temperature and salinity rise, the velocity of sound increases. As the pressure increases due to the depth, the velocity of sound also increases. These variances are not severe but when the measurements are in tenths of a second, any change will cause inaccuracies.

FIGURE 4-9

WATER TEMPERATURE PROBLEMS

type ==> Continue, Return



When arrays are positioned, as in the above display, it allows the system to cover a large area of ocean. In this example an area 400 miles by 400 miles or roughly the size of Montana. This area can be further subdivided into 100 sectors, each 40 miles square to make the tracking easier. Each of these sectors can presently be covered by XXX correlations per hour.

FIGURE 4-10

AREA BREAKDOWN

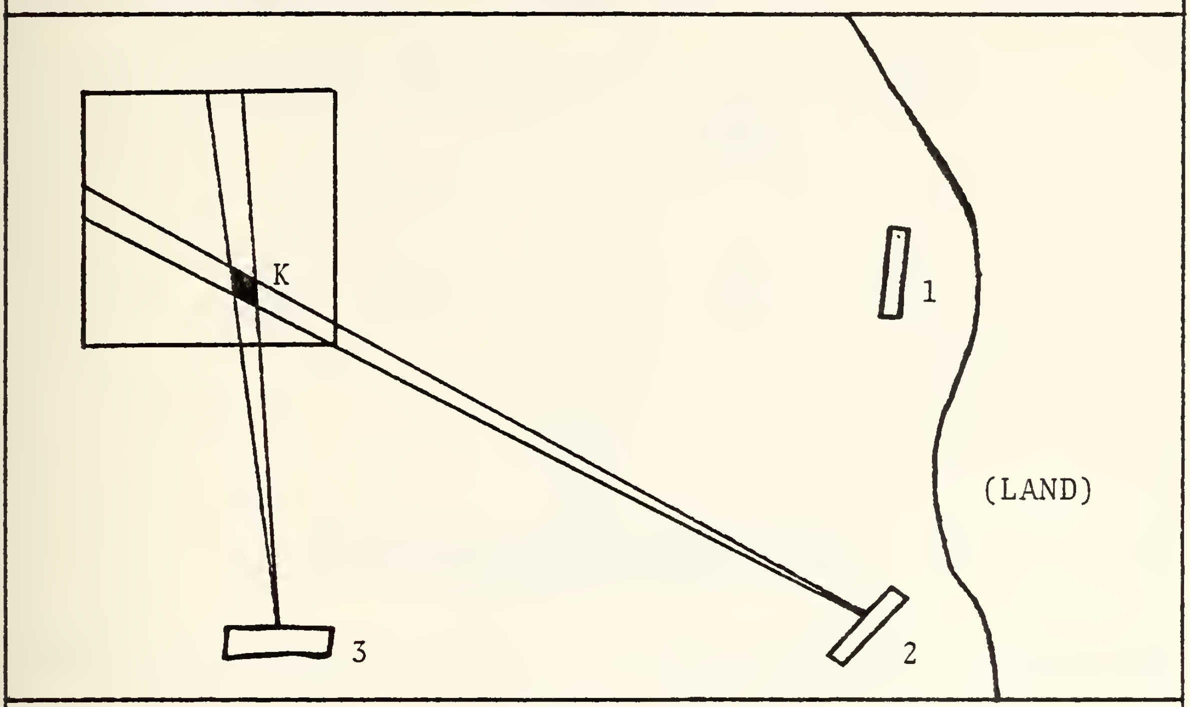
E. HYDROPHONES

The next portion of the tutorial provides an explanation of the hydrophone, the arrays, and steering the arrays toward the area of surveillance. Figures 4-11 through 4-13 illustrate the basic principles of correlation and the characteristics of the hydrophone arrays. As figure 4-11 exhibits, the area covered by a single beam increases with distance from the array and accuracy decreases proportionately.

The hydrophone array is a long tube, about the size of a large fire hose in circumference and roughly 2,000 feet long, containing approximately 60 hydrophones. Each hydrophone works in the same manner as a sound speaker in reverse, it receives the waves of sound and converts them into electrical signals. Figure 4-12 illustrates the rough shape of a beam formed by the hydrophone array. This shape varies in width and length with frequency, a high frequency beam, for example, being short and narrow.

As figure 4-12 illustrates that the beam is directed straight out perpendicular to the array when no corrections are made to it. However, it can be directed as figure 4-13 illustrates. The shifting of the direction is accomplished by delaying the signal received by some of the hydrophones longer than others. Since there are about 60 hydrophones, there can be about 60 possible combinations producing roughly 60 beams to cover the area.

type ==> Continue, Hydrophones, Ellipsoids, Return

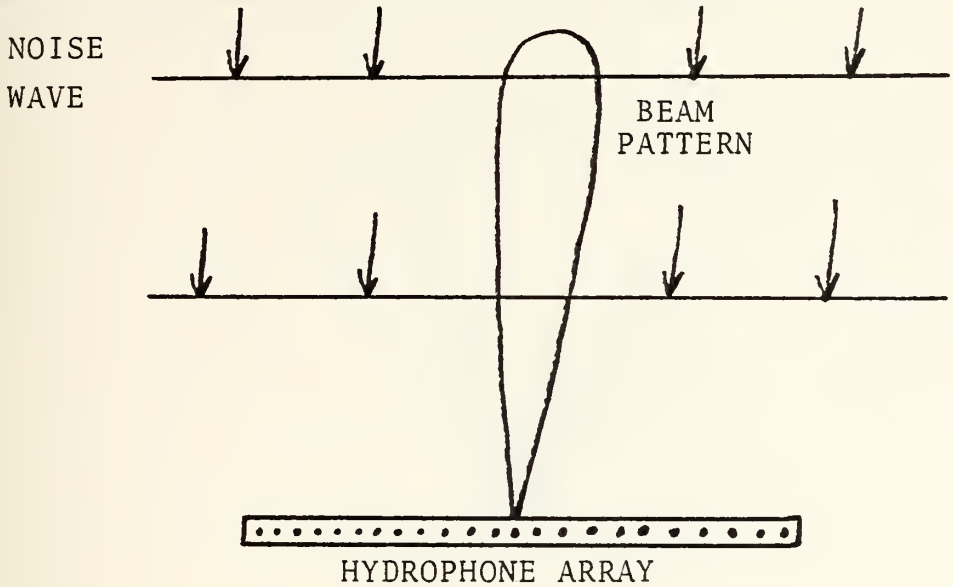


Correlation is the process of determining the relationship of signals received at two different points. Arrays 2 and 3 receive signals continuously, process them by time-delay techniques, and analyze the results to determine the degree of correlation. When the correlation level is significantly high, then a possible contact is recorded. A series of contacts establishes a track with various possible levels of certainty.

FIGURE 4-11

CORRELATION

type ==> Continue, Array (pointing), Ellipses, Return

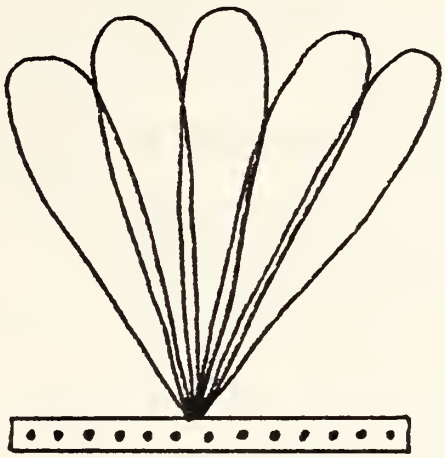


The system for collecting signals is an array of hydrophones, roughly 2000 feet long containing 60 passive sonar devices. If no adjusting (pointing) of the array is done, then only noise waves coming in directly abeam of the array will be detected. The noise arrives at the array in the same manner that a wave crashes on the beach, some come in straight and others hit at an angle.

FIGURE 4-12

HYDROPHONES

type ==> Continue, Hydrophones, Ellipses, Return



In order to cover a large area with a single pair of arrays, the beams must be able to be pointed in several directions. This is accomplished by using a time shift in the receiving of a signal at various hydrophones along the array. There can be roughly as many beam patterns as there are hydrophones and the beam patterns are generally overlapped to ensure that the entire area is covered.

FIGURE 4-13

ARRAY POINTING

F. ELLIPSES

When contact is made by an array, the area of probable contact is a very long narrow ellipse centered along a tau line. When two different arrays detect a signal then the ellipse is centered at the intersection of the tau lines, and its shape is dependent on their angle of intersection. As the display (figure 4-14) shows, when the angle of intersection of the tau lines is nearly perpendicular, the ellipse approaches a circle and therefore there is a much smaller area of probable contact. As the tau lines become more parallel the ellipse becomes longer and the probable contact area increases. When more data is accumulated on the signal, additional ellipses can be drawn as the display (figure 4-15) demonstrates. The additional ellipses are nearly totally confined within the previous ellipses, thus reducing the probable contact area even further. If the data received alternates between different sets of arrays, as the first three ellipses illustrate, then the area of probable contact shrinks rapidly. If the data comes from the same set of arrays then the ellipse becomes longer and narrower and eventually exceeds prior ellipses in total area.

The display (figure 4-15) demonstrates how graphics can animate displays to make them appear as if they were being drawn by an instructor. When the original display is presented, only the two outermost ellipses are drawn in the middle section. Then as the user finishes reading the explanation of combining ellipses he has the option of

seeing more ellipses combined by typing 'm' or continuing the tutorial by typing 'c'. If more ellipses are desired they will be drawn one at a time with a two second delay between them. Another feature of this display is that it can be reached by two different routes. First it will be displayed when the user requests more detail in earlier displays (figures 4-11 and 4-14) or by following the basic simplified route. Whichever route is followed this display will occur at least once.

G. ENTIRE SYSTEM

Now that the user has seen what the system is trying to accomplish, a simplified view of the complexity of the system is displayed (figure 4-16). This display and its detailed subsystems prepare the individual for the actual use of the system.

If the user decides to probe into the inner workings of the system, he has the opportunity at this point to see the basic technique and some of the formulas used in solving the correlation. The first display (figure 4-17) describes the use of coherent versus incoherent calculations. This display demonstrates that straight text can be inserted into the tutorial: however this is not a very cost effective utilization of the computer. The second display (figure 4-18) illustrates the formulas that are used in computing the correlation. This display demonstrates that formulas, though difficult to display, can be presented in a clear and complete manner. Again this is not an efficient method of

communicating this information, but it does demonstrate that it can be done.

Some of the variables used in the calculations can be varied by the tracker to obtain a better picture of the actual situation. This display (figure 4-19) provides the user with just a few of the possible variables and their upper and lower limits. If during the scenario or actual system utilization, the tracker attempts to exceed these limits the computer will not allow him to input the changes.

type==> Continue, Ellipses (combined), Hydrophones, Return

DIAGRAM A

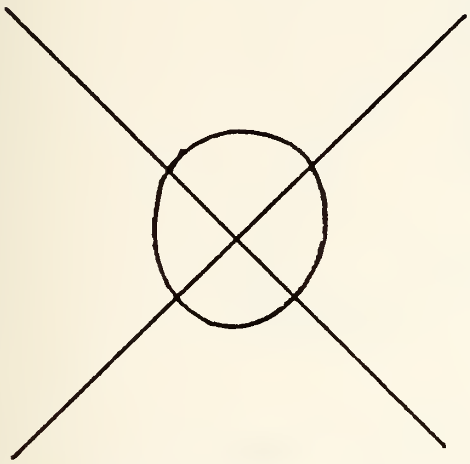
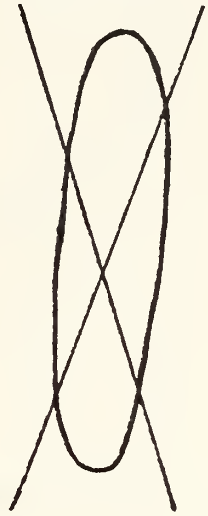


DIAGRAM B

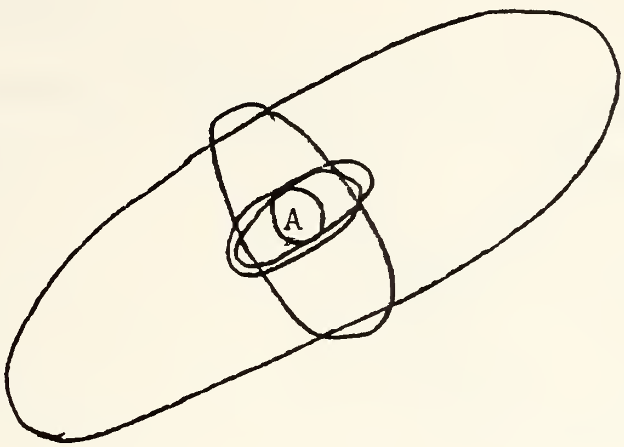


When a correlation is made from data received at two arrays, an area of probable contact is established. The area of probable contact forms an ellipse, as in diagrams A and B. When the tau lines are nearly perpendicular, the ellipse approximates a circle which indicates a high probability of contact within a small geographic area. However, if the tau lines intersect at relatively small angles, the result is a very long ellipse, indicating a lower probability of contact within the area.

FIGURE 4-14

ELLIPSES

type ==> Continue, Ellipses, Hydrophones, Return



Area around A is now the probable contact area.

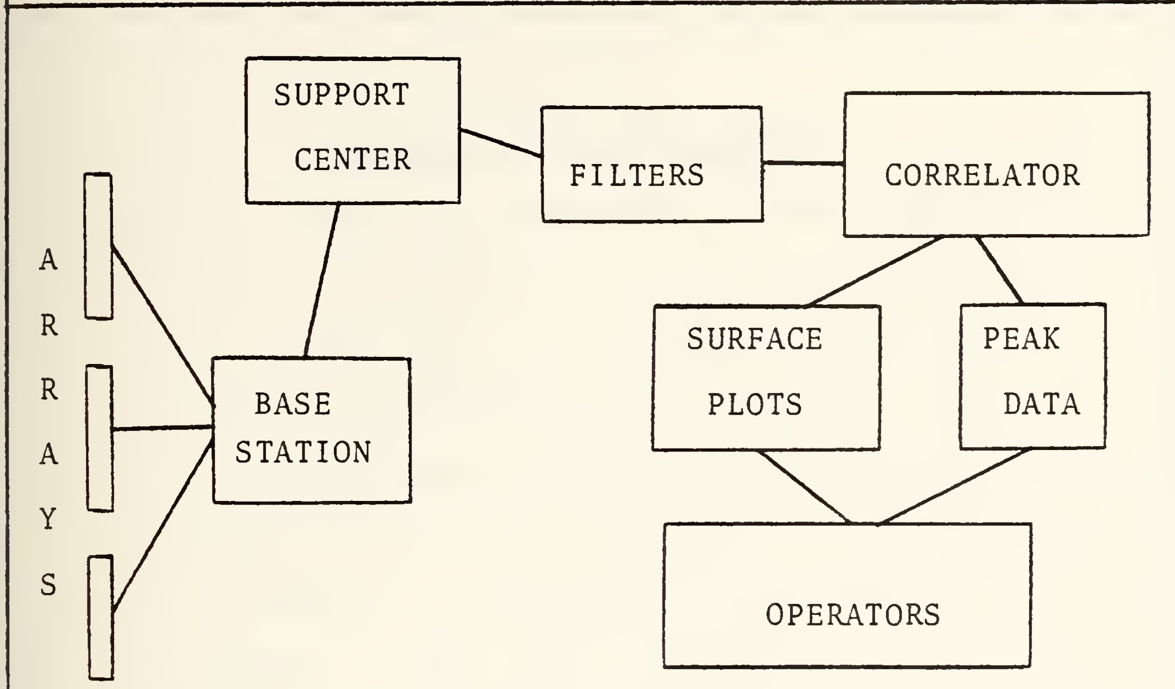
When several pieces of data are gathered about a specific signal, then the ellipses can be combined to reduce the area of probable contact. As the diagram illustrates, the second set of data forms an ellipse which is inside the first ellipse. Additional data can be used to establish additional ellipses and eventually improve the detection of targets.

Type 'm' to see how additional ellipses are combined.

FIGURE 4-15

COMBINED ELLIPSES

type==> Continue, Techniques (correlation), Filters, Return



A brief overview of the entire system demonstrates the amount of complex machinery required to accomplish this surveillance. Signals are received at the arrays and are transmitted to the base station, then sent via satalite to the Support Center (SC). In the SC the data is sent through filters and a correlator, where the computer establishes two sets of data (peak data files and surface plots). The peak data and surface plots are reviewed by the operators to determine any possible target tracks.

FIGURE 4-16

ENTIRE SYSTEM

type ==> Continue, Rules (formulas), Filters, Return

STATEMENT OF THE TECHNIQUE

Given two sequences of time series data, a function represents the degree of match of the two time series as a function of adjustment of the delay and of adjustment of the relative Doppler ratio between the two series. The term ''coherent'', applied to this function, implies that the relative phase evolutions of the time series are used in determining the degree of match, rather than discarded as in an incoherent calculation. The time series of interest are finite, discrete, frequency-shifted, band-limited, complex time series obtained from passive sonar sensors (hydrophone arrays).

FIGURE 4-17

CORRELATION TECHNIQUE

type ==> Continue, Filters, Return

1) $$\left|X(a,t)\right| = \frac{1}{A}\left|\sum_{n=0}^{N-1} X_1'(n)\bullet X_2^*(n-t/\Delta T)e^{2\pi i(aF1-F2-n\ T}\right|$$

2) $X_1'(n)=X_1(an-(a-1)N/2)$

3) $$A^2=\sum_{n=0}^{N-1}X_1'(n)\bullet X_1^*(n)\sum_{m=0}^{N-1}X_2(m-t/\ T)X_2^*(m-t/\ T)$$

where X(a,t) is the modulus correlation coefficient
a is the relative doppler ratio compensation
t is the delay compensation
A is the normalization constant
X_1 and X_2 are time series sampled at time intervals T
F1 and F2 are the center frequency of the passband for the time series

The basic correlation structure can be seen in equation 1, as the modulus of normalized sum of products of the time series samples (with one series conjugated). The other two equations are used for delay and Doppler compensation of the time series prior to correlation.

FIGURE 4-18

CORRELATION FORMULAS

type ==> Continue. Return

	RANGES
LONGITUDE	30.00-50.00
LATITUDE	150.00-130.00
COHERENCE LIMITS	0.0-2.0
TRACK NUMBER	1-10
RUN NUMBER	1-7
REGION NUMBER	1-25

Listed above are a few of the parameters that can be adjusted to get a better track of a vessel. Other parameters are demonstrated throughout the tutorial and if the limits are exceeded, the computer will adjust and notify the operator.

FIGURE 4-19

PARAMETER VARIABLES

V. DATA DISPLAY TUTORIAL

This section of the tutorial is designed to prepare a tracker for the types of data that will be presented during operational conditions. It provides some representative samples of data in order to explain explicit details about the values in certain categories. A review of this part of the tutorial would be advisable for all trackers as a routine before each watch they stand because it provides many of the basics of each data representation. Such a review would ensure that the operator will not totally forget characteristics of the system during periods of disuse.

This section of the tutorial begins with an introductory display (figure 5-1) which presents the basic concepts of this section and then describes the importance of a thorough understanding of each data display. None of the data displays can be overlooked by a tracker since the data is interconnected in such a manner that the accuracy of one forms a check on another. For example, the values of the position-data display (figure 5-2) are checked for variance by the values of the covariance matricies in figure 5-7. This interconnection is useful in obtaining a good overall picture of the situation but still allows the user the freedom to transverse the data displays in any desired order.

DATA TUTORIAL INTRODUCTION

This section of the tutorial deals with the types of data that will be displayed when a tracker is utilizing the actual operational system. The displays give representative samples of data and then explain the characteristics and point out key points. This section of the tutorial is brief but extremely important to the understanding of the operator's job. A quick review of this tutorial would be beneficial to the experienced operator before starting each session on the operational system.

type c to continue

FIGURE 5-1

DATA TUTORIAL INTRODUCTION

The positional data display (figure 5-2) provides the tracker with basic information about the vessel's probable position, course, and speed. The latitude and longitude will not vary much whether the signals are from a target or noise; however, they can be used to determine a good starting point for setting the parameters for a surface plot. The courses and speeds will vary depending on the type of signal that is being received and thus should be closely examined. For an experienced operator, the covariance matrix display (figure 5-7) can be quite useful in determining the validity of a suspected track.

The peak data display (figure 5-3) provides the operator with some rather difficult numbers to comprehend. The STM/BM numbers are the numerical representation of the array and the direction of the beam. It is important to recognize when a different array is receiving the signal because this can increase the accuracy of triangulation. The tau values are important because they will vary in the same manner as the courses and speeds; however, this variance can only be detected when using the same pair of arrays. A large jump in tau values, as in figure 5-3, is another indication that two different pairs of arrays are detecting the same signal, but the variance in values from a single pair of arrays is quite good. The doppler values will also vary and should be watched closely when tracking a surface vessel.

type ==> Continue, Return

RUN NUMBER 6 REGION NUMBER 6 TRACK NUMBER 9

Water Time 23-Feb-80	Peak Number	Position Coordinates Latitude	Longitude	Courses and Speeds Course	Speed
14:58:03	0	39.5000	147.5000	0.0	0.00
14:58:03	3427	39.4711	147.4707	142.0	5.39
15:04:23	3533	39.4682	147.4552	154.7	5.52
15:14:30	3449	39.4212	147.4189	153.9	5.96
15:38:35	3540	39.3582	147.4417	160.3	6.09
15:48:44	3452	39.3980	147.4815	161.6	5.59

This data illustrates the estimated position, course and speed of a possible target. When the data is relatively constant, as in the above display, it is probable that the signals were generated by a vessel. If the courses were 159.0, 135.6, 160.4, 122.7, 150.3 then the data would probably be due to extraneous noise.

FIGURE 5-2

POSITION-DATA

type ==> Continue, Return

RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9

Peak	-Ref Station-		-PRD Station-		----Peak Data----	
Number	STM/BM	Freq	STM/BM	Freq	TAU	DOPPLER
Ø	Ø	Ø.ØØ	Ø	Ø.ØØ	Ø.Ø	.ØØØØ
3427	1323Ø24	41.4Ø	1322Ø29	41.4Ø	-416.7	-.ØØØØ
3533	1322Ø31	41.4Ø	2411Ø2Ø	41.4Ø	27.3	.ØØØØ
3449	1323Ø24	41.4Ø	1322Ø3Ø	41.4Ø	-412.8	-.ØØØØ
354Ø	1322Ø31	41.4Ø	2411Ø21	41.4Ø	19.2	.ØØØØ
3453	1323Ø25	41.4Ø	1322Ø3Ø	41.4Ø	-42Ø.7	-.ØØØØ

This data display presents the peak numbers, the array that received the signals, the frequency of the signal, and peak data. The variances in the Tau and Doppler values can give evidence as to the validity of the track. The variance in the Tau values from -416.7 to 27.3 to -412.8 etc. means that the ellipses will probably combine very well because the data is being collected from two different sets of arrays.

FIGURE 5-3

PEAK-DATA

The accuracy data display (figure 5-4) is a good indication of track validity because the Chi-Square probability scores describe the consistency of the received data. The more data received about a certain track, the better the exact location can be determined. The cumulative Chi-Square score is an excellent quick reference because the score is constantly normalized to present a true indication of the probability. It does not matter how many scores are obtained because they are all averaged. The cumulative score will approach one (1.0) if the signals being received are from a target.

The surface plots (figures 5-5 and 5-6) represents the signals received in another fashion. They are compiled by taking all of the data received by array pairs and then filtering the data through a coherence filter that removes all signal data below the level set. Then the data is displayed for the region that the tracker has set for surveillance. Both the coherence limit and the regional area are set by the operator in the attempt to locate a target.

The last display in this section of the tutorial is the covariance matrix display (figure 5-7) which illustrates the complexity of the system as it presently stands. It takes either a mathematician or a very experienced tracker to evaluate the matrices displayed. For this reason an algorithm is being developed by a Navy contractor to simplify this data. However, at this time,

it is still important to the evaluation of the previous data, and is being displayed in this fashion until the new algorithm is completed.

type ==> Continue, Return

RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9

Peak Number	Prob Score	--CHI Square Scores-- Stagewise	Cumulative	Meas Residuals Deltat	Deltaf
0	0.00	0.0000	0.0000	0.0000	0.0000
3427	-1.80	0.3171	0.3171	1.7978	-0.0405
3533	-4.89	0.0162	0.1667	0.6844	0.0193
3449	-3.99	0.4410	0.2581	3.3146	-0.0036
3540	-4.66	1.4096	0.2460	-7.4522	0.0114
3453	-4.81	2.0796	0.8527	-8.3141	0.0043
3421	-4.31	1.8945	0.8892	-6.4137	0.0125

This data shows the same peaks, their Chi Square probabilities of actual position, and their measured residuals. The Chi Square scores are broken into two groups, individual scores and the cumulative total. The cumulative score is a good check as to the probability of an actual contact, scores near 1.0 are quite good.

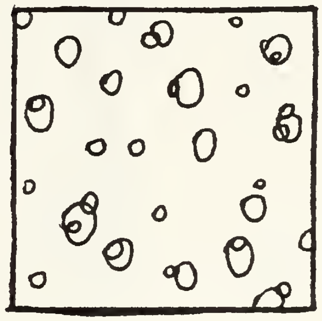
FIGURE 5-4

ACCURACY-DATA

type ==> Continue, Samples, Return

REGION NUMBER 8

COHERENCE LIMIT 1.0



By examining a surface plot the operator may be able to recognize a pattern of peaks that would establish a target's track. Care must be taken when viewing the surface plots because the size of the area (range of latitude and longitude) can be varied, thus giving the appearance of a track when there might not be one.

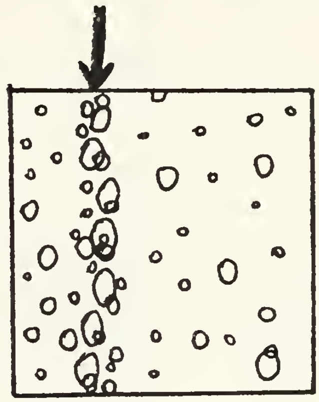
FIGURE 5-5

SURFACE-PLOT

type ==> Continue, Samples, Return

REGION NUMBER 5

COHERENCE LIMIT 1.0



This surface plot shows a very probable track, indicated by the arrow. Because the size of the plot is relatively small (60 miles square) and the coherence limit is relatively high (1.0), this is probably a track of a target and should be further investigated.

FIGURE 5-6

SAMPLE SURFACE-PLOT

type ==> Senerios, Tutorial (again), Return

RUN NUMBER 6 REGION NUMBER 6 TRACK NUMBER 9

Peak Number	--------COVARIANCE	MATRIX--------		
3427	0.03136	0.00000	-0.03150	0.00000
	0.00000	37.88962	0.00000	-48.49765
	-0.03150	0.00000	0.07311	0.00000
	0.00000	-48.49765	0.00000	62.21477
3533	0.00985	0.00047	0.00551	-0.00001
	0.00047	0.31107	0.00178	0.19371
	0.00551	0.00178	0.00944	-0.00179
	-0.00001	0.19371	-0.00179	0.13686

This display illustrates the covariance matrices which can be used to determine the accuracy of the estimated courses and speeds. An experienced operator could determine this by examining each 4 X 4 matrix. However, this is difficult and time consuming. Efforts are being made to have this handled by the computer.

FIGURE 5-7

COVARIANCE MATRICES

VI. SCENARIO

A. PURPOSE

The scenario is an important part of the learning experience offered in this tutorial because it offers the user the opportunity to practice the skills that have been taught. Parameters can be inserted and the data displayed as it would actually be shown on the tactical system and also advice could be given as to the quality of the data. This actual hands-on training has many benefits over conventional methods. It reduces the anxiety of the user when finally confronted with a tactical mission, gives him key points that can be expanded in future lessons, refresher training at his work place, and a wide variety of options to try on tactical data outside a mission.

A great deal of anxiety and frustration generally is experienced by the individual when he starts a new endeavor. Any reduction of this anxiety would be beneficial to the performance of the individual and thus improve his value as an operator. With an embedded tutorial an individual can learn the workings of the system and the procedures of the task while becoming familiar with the computer operations. This scenario is designed to give the user several different looks at the kind of data and displays that he will be working with every day. A tutorial on the actual system would be far

better than the mock-up used in this study because the using of actual data would enable the user to see many more combinations of parameters. This tutorial would be altered to analyze the parameters more carefully to establish what comments would be appropriate for the data that is displayed.

The scenario presently has only four sets of data and explanations have been limited by the data available and the size of the computer. Since the tactical system receives the data directly from its storage bank, both of these limitations are removed. The tutorial program would only have to receive the data, analyze it, and make appropriate comments. These comments, as in the sample scenario, would describe to the user the areas to watch in order to get the best track, and also it would help in making parameter settings for optimum surveillance. These key points would be beneficial to both the new user and as a refresher for the the experienced operator.

Since the scenario is embedded in the actual tracking system, it is easy to have the operators run refresher training on tactical data outside an actual mission, without sending them out of the facility. This training could increase the capabilities of an experienced operator because it would serve as a reminder of some option areas that might not normally be utilized either because they are unfamiliar or because they were not very effective on the data that has been previously tried. Since each new

track is unique in its properties, an operator must be very flexible in his attempt to distinguish a track of a target from noise.

As the data is tactical but not part of a current operation during a senario, it is an excellent time for an experienced operator to try various combinations of parameters in the attempt to track a target more accurately. An operator would not have this option during operation due to the necessity of constant surveillance of the area.

B. PROGRAM SETUP

The scenario program is written to allow the expansion of this small sample scenario with little modification. The program takes the input of the user and then displays the data formatted in the manner that the user desires. As this demonstration program is limited in data available, only four possible sets of displays can be shown, each dependent on the parameters inserted by the user. The possibilities of displays range from excellent parameters to worst-case parameters with two fair sets of data displays in the middle. In each set of displays the user can see peak data, location data, accuracy data, and surface plots of the region. The user is also given the opportunity to display the parameters that are presently being used and given an opportunity to alter any or all of them. Figure 6-1 shows the format of the parameters that can be altered by the user to improve surveillance of an

area. The present values shown in the figure are the parameters that the system initialized when the scenario was entered. These parameters will give the best-case set of displays and explanations, but they can be altered to demonstrate the effect of poor parameters or to highlight specific parts of certain data displays. As was mentioned earilier there are no set limits or firm parameters that will give the best-case display of data in all situations.

C. DISPLAY EXPLANATION

There are four sets of displays: 1) best-case parameter setting, 2) data accumulated over too large an area, 3) data accumulated with too low a coherence limit, and 4) worst-case data which was accumulated over too large an area at too low a coherence limit. These displays will be grouped by display category to demonstrate the pitfalls in each type of display for certain choices of parameters.

1. Positional-Data

The positional data is probably the best source of information that the operator can obtain about a possible target because it provides a smaller area to search and a good indication of the probability of a target. Furthermore it is the easiest set of data to understand because it is given in terms that most people can easily grasp, position with a course and a speed. Some of the other displays present tau lines, doppler, and cumulative Chi-Scuare values. The positional data displays (figures

6-2 to 6-5) present the peak numbers, calculated position, course, and speed of a suspected target. A key point to observe is the variance in the course or speed because this would show that the track is not a firm set of data and that it could be noise.

As figure 6-2 illustrates, the courses and speeds are quite steady and well within reason; the speeds are not excessive and the courses are roughly in the same direction. In contrast, figure 6-5 has courses that vary 30 degrees, from 005.3 to 034.6, and speeds that jump from 7.69 to 4.69 causing doubt that this is an actual track because vessels do not generally proceed in this fashion. These were two extremes, excellent data and worst-case data; however, it is difficult to determine the validity of the track for most data. Figures 6-3 and 6-4 show some more common data in which the courses and speeds vary somewhat but they are gradual changes rather than abrupt changes.

```
type ==>  the desired number change,  Contirue,  Return
```

```
To change any of the present values of the parameters to
new values, use the following format:
          PARAMETEP-NUMBER, NEW-VALUE n
For example ==> 4,1.2n changes the coherence limit to 1.2

                    MENU OF PARAMETERS

    LIMIT          PARAMETER        PRESENT VALUE   NEW VALUE

    1-25         1. REGION               5
    1-7            2. RUN                6
    1-10          3. TRACK               9
  0.1-2.0       4. COHERENCE             1.0
 30.00-50.00    5. LATITUDE(high)       35.50
 30.00-50.00    6. LATITUDE(low)        36.00
120.00-140.00   7. LONGITUDE(high)     124.50
120.00-140.00   8. LONGITUDE(low)      125.00
```

FIGURE 6-1

MENU FOR PARAMETER CHANGES

RUN NUMBER 6 REGION NUMBER 6 TRACK NUMBER 9

Peak Number	Position Coordinates Latitude	Longitude	Courses and Speeds Course	Speed
0	39.5000	147.5000	0.0	0.00
3427	39.4711	147.4707	142.0	5.39
3533	39.4682	147.4552	154.7	5.52
3449	39.4212	147.4189	153.9	5.96
3540	39.3582	147.4417	160.3	6.09
3453	39.3980	147.4815	161.6	5.59
3421	39.4021	147.4739	162.4	5.82
3479	39.4219	147.4632	160.3	5.71
3429	39.4037	147.4297	161.1	5.59

A VERY GOOD TRACK WITH STEADY COURSES AND SPEEDS.

FIGURE 6-2

EXCELLENT POSITIONAL-DATA

RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9

Peak Number	Position Coordinates Latitude	Longitude	Courses and Speeds Course	Speed
0	38.5000	142.5000	0.0	0.00
3542	38.4731	142.4873	142.0	6.81
3478	38.4422	142.4791	150.6	6.72
3548	38.4431	142.4762	147.3	6.04
3557	38.4401	142.4750	156.4	6.87
3461	38.4324	142.4775	156.8	6.94
3640	38.4246	142.4689	144.7	6.42
3582	38.4263	142.4663	136.2	6.75
3533	38.4221	142.4572	149.7	6.35

POSSIBLY A GOOD TRACK, BUT THE SIZE OF THE REGION BEING COVERED IS TOO LARGE.

FIGURE 6-3

LARGE AREA POSITIONAL-DATA

RUN NUMBER 1   REGION NUMBER 3   TRACK NUMBER 4

Peak	Position Coordinates		Courses and Speeds	
Number	Latitude	Longitude	Course	Speed
0	33.5000	145.5000	0.0	0.00
3325	33.4734	145.5398	228.4	7.34
3342	33.4689	145.5402	235.6	9.27
3394	33.4502	145.5693	243.5	10.21
3327	33.4294	145.5728	227.9	8.38
3319	33.3825	145.5829	222.5	7.03
3364	33.3518	145.5962	225.3	7.86
3318	33.3329	145.6036	233.1	8.37
3343	33.2185	145.6285	242.3	7.89

EVEN THOUGH THE COURSES ARE FAIRLY STEADY, THE SPEEDS ARE FLUCTUATING GREATLY DUE TO THE LOW COHERENCE LIMIT.

FIGURE 6-4

LOW COHERENCE POSITION-DATA

RUN NUMBER 3   REGION NUMBER 4   TRACK NUMBER 7

Peak Number	Position Coordinates Latitude	Longitude	Courses and Speeds Course	Speed
0	36.5000	143.5000	0.0	0.00
3560	36.4784	143.6534	020.5	4.58
3532	36.5348	143.5739	034.6	6.27
3571	36.5782	143.4936	025.9	6.02
3564	36.6139	143.4467	012.3	7.69
3551	36.6317	143.4052	005.3	4.69
3554	36.6308	143.3761	014.7	6.34
3584	36.6536	143.3485	018.1	4.25
3547	36.6738	143.3263	023.5	5.28

POOR TRACK AS SEEN BY THE COURSE AND SPEED CHANGES, PROBABLY DUE TO THE LARGE AREA AND THE LOW COHERENCE LEVEL.

FIGURE 6-5

WORST CASE POSITION-DATA

2. Peak-Data

The peak data displays (figures 6-6 to 6-9) exhibit some rather confusing data in that most people know little about tau lines and fewer people have any concept of numbers associated with doppler. The STM/BM numbers are a numerical description of the array and the direction in which the beam is pointing when it received the signal. A good track will have alternating STM/BM numbers, depicting several arrays picking up the data and thus the triangulation will be more precise. Another place to observe the alternating of arrays is the tau values. However care must be taken not to confuse the varying of the numbers with the alternating of arrays. If the track is weak or the data is from noise it could appear to be alternating when it is actually just varying a great deal. The tau values obtained from a single array should not vary a great deal between peaks. When tracking a surface vessel the doppler values will be very steady and the tau values will fluctuate a little more than for a submarine.

Figure 6-6 illustrates a pattern of constantly switching array pairs in the reception of data about the track, while any data received by a single set of arrays does not vary a great deal. For example the switching of tau values from -416.7 to 27.3 to -412.8 to 19.2 shows that two different sets of arrays are picking up the signal, while -416.7, -412.8, -420.7 of a single array pair show that the values are fairly steady. In contrast

figure 6-9 illustrates that signals are only being received by one set of arrays, that no switching occurs, and that while in a single array pair the tau values vary quite extensively, 62.1 to 25.7 in a short period. Again these are somewhat extreme cases and are easy to interpret compared to the data displayed in figures 6-7 and 6-8, where there is some switching and some fairly steady readings.

3. Accuracy-Data

The accuracy data displays (figures 6-10 to 6-13) illustrate some data that is good for determination of the quality of a track since it provides the operator with the Chi-Square probability of a target being within a specific region, generally an ellipse. The individual scores are important to see that there are no drastic jumps that would adversely affect the cumulative score. For instance, a large single individual score could greatly increase the cumulative score and present a false impression of a good track. The important scores to watch are the cumulative totals which should approach one if it is a definite track of a target.

Figure 6-10 displays the pattern of a good Chi-Square accumulation in that the individual scores are constantly getting better and the cumulative score is approaching one. The individual scores are getting higher and higher because the probability ellipses are combining and thus reducing the size of the probable contact area,

figure 4-17. In the worst-case accuracy data display (figure 6-13), the individual scores do not appreciably increase because all the data is being received at one array pair and the ellipses are combining in such a manner that the area of probable contact does not significantly shrink.

RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9

Peak	-REF Station-		-PRD Station-		---Peak Data---	
Number	STM/BM	Freq	STM/BM	Freq	TAU	DOPPLER
Ø	Ø	Ø.ØØ	Ø	Ø.ØØ	Ø.Ø	.ØØØØ
3427	1323Ø24	48.7Ø	1322Ø29	48.7Ø	-416.7	-.Ø4Ø5
3533	1322Ø31	48.7Ø	2411Ø2Ø	41.ØØ	27.3	.Ø117
3449	1323Ø24	48.7Ø	1322Ø3Ø	48.7Ø	-412.8	-.Ø444
354Ø	1322Ø31	48.7Ø	2411Ø21	48.7Ø	19.2	.Ø229
3453	1323Ø25	48.7Ø	1322Ø3Ø	48.7Ø	-42Ø.7	-.Ø4ØØ
3421	1322Ø32	48.7Ø	2411Ø2Ø	48.7Ø	22.7	.Ø245
3479	1323Ø24	48.7Ø	1322Ø31	48.7Ø	-418.6	-.Ø437
3429	1322Ø31	48.7Ø	2411Ø23	48.7Ø	25.1	.Ø235

GOOD TRACK WITH TWO SETS OF ARRAYS SWITCHING BACK AND FORTH TO PROVIDE DATA. (NOTICE THE TAU VALUES ALTERNATING)

FIGURE 6-6

EXCELLENT PEAK-DATA

RUN NUMBER 3   REGION NUMBER 4   TRACK NUMBER 7

Peak	-REF Station-		-PRD Station-		---Peak Data---	
Number	STM/BM	Freq	STM/BM	Freq	TAU	DOPPLER
0	0	48.70	0	48.70	0.0	.0000
3542	1333031	48.70	1346024	48.70	19.4	.0201
3478	1333041	48.70	1346031	48.70	12.3	.0237
3548	1333027	48.70	1346028	48.70	16.4	.0394
3557	1322107	48.70	2411321	48.70	-426.3	.1027
3461	1333034	48.70	1346021	48.70	27.8	.0384
3640	1333051	48.70	1346039	48.70	18.2	.0392
3582	1322114	48.70	2411324	48.70	-440.4	.1379
3533	1333029	48.70	1346042	48.70	11.4	.0369

NOT MUCH ALTERNATING OF ARRAY PAIRS, PROBABLY NOT TOO GOOD A TRACK.

FIGURE 6-7

LARGE AREA PEAK-DATA

RUN NUMBER 1 REGION NUMBER 3 TRACK NUMBER 4

Peak	-REF Station-		-PRD Station-		---Peak Data---	
Number	STM/BM	Freq	STM/BM	Freq	TAU	DOPPLER
0	0	48.70	0	48.70	0.0	.0000
3325	1341021	48.70	1472365	48.70	22.7	.0391
3342	1834261	48.70	2437001	48.70	-312.4	-.0391
3394	1341036	48.70	1472341	48.70	15.6	.0488
3327	1834275	48.70	2437009	48.70	-328.4	-.0352
3319	1341028	48.70	1472355	48.70	28.4	.0592
3364	1341025	48.70	1472351	48.70	39.7	.0491
3318	1834268	48.70	2437005	48.70	-321.1	-.0379
3343	1341029	48.70	1472368	48.70	21.8	.0529

FAIRLY GOOD SWITCHING OF ARRAYS BUT WITHIN AN ARRAY PAIR THERE IS TOO MUCH VARIANCE DUE TO THE LOW COHERENCE LIMITS.

FIGURE 6-8

LOW COHERENCE PEAK-DATA

RUN NUMBER 5 REGION NUMBER 5 TRACK NUMBER 22

Peak	-REF Station-		-PRD Station-		---Peak Data---	
Number	STM/BM	Freq	STM/BM	Freq	TAU	DOPPLER
0	0	48.70	0	48.70	0.0	.0000
3560	2411031	48.70	1441024	48.70	48.3	.0243
3532	2412034	48.70	1442067	48.70	62.1	.0294
3571	2411041	48.70	1441028	48.70	36.2	.0362
3564	2411038	48.70	1441021	48.70	25.7	.0329
3551	2411036	48.70	1441025	48.70	28.3	.0338
3554	2411028	48.70	1441020	48.70	49.6	.0195
3584	2411031	48.70	1441024	48.70	41.2	.0293
3547	2411035	48.70	1441018	48.70	55.7	.0264

NO SWITCHING OF ARRAY PAIRS, BUT A LOT OF VARIANCE IN THE TAU VALUES. A VERY POOR TRACK.

FIGURE 6-9

WORST CASE PEAK-DATA

RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9

Water Time	peak	Prob	--CHI Square Scores--	
23-Feb-80	Number	Score	Stagewise	Cumulative
14:58:03	0	0.00	0.0000	0.0000
14:58:03	3427	-1.80	0.3171	0.3171
15:04:23	3533	-4.89	0.0162	0.1667
15:14:30	3449	-3.99	0.4410	0.2581
15:38:35	3540	-4.66	1.4096	0.2460
15:48:44	3453	-4.81	2.0796	0.8527
15:49:31	3421	-4.31	1.8945	0.8892
15:50:56	3479	-4.75	2.0038	0.9327
15:52:34	3429	-4.89	2.2893	0.9402

OUTSTANDING CUMULATIVE SCORE BY THE FINAL TIME PERIOD,
A VERY GOOD TRACK.

FIGURE 6-10

EXCELLENT ACCURACY-DATA

RUN NUMBER 3 REGION NUMBER 1 TRACK NUMBER 4

Water Time 23-Feb-80	Peak Number	Prob Score	-CHI Square Scores-- Stagewise	Cumulative
16:52:56	0	0.00	0.0000	0.0000
16:52:56	3542	-1.49	0.1435	0.1435
16:55:37	3478	-1.27	0.2159	0.1537
16:58:10	3548	-1.82	0.5382	0.3829
16:59:19	3557	-1.95	0.6547	0.4690
17:03:02	3461	-2.56	1.1021	0.7023
17:05:56	3640	-1.13	0.2371	0.6501
17:09:42	3582	-2.03	0.6829	0.6928
17:11:48	3533	-3.05	0.7928	0.7520

THE CUMULATIVE SCORE IS FAIR BUT NONE OF THE INDIVIDUAL SCORES IS VERY HIGH.

FIGURE 6-11

LARGE AREA ACCURACY-DATA

RUN NUMBER 2 REGION NUMBER 3 TRACK NUMBER 6

Water Time 23-Feb-80	Peak Number	Prob Score	--CHI Square Scores-- Stagewise	Cumulative
06:12:37	0	0.00	0.0000	0.0000
06:12:37	3325	-0.83	0.2134	0.2134
06:13:25	3342	-1.52	0.4327	0.3463
06:14:03	3394	-1.83	0.5621	0.4743
06:14:57	3327	-1.97	0.7631	0.6411
06:16:32	3319	-2.24	0.6428	0.6229
06:17:29	3364	-2.56	0.8817	0.7536
06:18:54	3318	-2.72	0.9443	0.7836
06:20:21	3343	-2.74	0.9218	0.7721

FAIR CUMULATIVE VALUES, BUT LOW INDIVIDUAL VALUES DUE TO LOW COHERENCE LIMITS.

FIGURE 6-12

LOW COHERENCE ACCURACY-DATA

RUN NUMBER 6   REGION NUMBER 4   TRACK NUMBER 7

Water Time 23-Feb-80	Peak Number	Prob Score	--CHI Square Scores-- Stagewise	Cumulative
21:36:04	0	0.00	0.0000	0.0000
21:36:04	3560	-0.90	0.1021	0.1021
21:37:09	3532	-1.43	0.2341	0.1735
21:37:52	3571	-1.78	0.3417	0.2638
21:38:38	3564	-1.93	0.3176	0.4173
21:39:41	3551	-2.06	0.3814	0.4328
21:40:02	3554	-2.54	0.4027	0.4458
21:40:58	3584	-2.36	0.3311	0.4637
21:41:44	3547	-2.21	0.3517	0.4528

VERY LOW CUMULATIVE SCORE. A VERY POOR TRACK.

FIGURE 6-13

WORST CASE ACCURACY-DATA

4. Surface-Plots

The surface plots can reveal a great deal about the actual situation in any area without picking any special track to correlate on because it displays the entire range of frequency values in that area. When the area is of proper size for the type of surveillance that is being conducted and the coherence limit is set high enough to reduce noise yet low enough to detect the required signal, then a surface plot can assist the operator in distinguishing a valid track.

Figure 6-14 illustrates the type of display that could help the operator to concentrate his efforts in the area on the left side of the plot. Care must be taken as to the size of the area and the level of the coherence limit because they could cause a surface plot to present the operator with a false impression of the actual situation, as figure 6-16 illustrates.

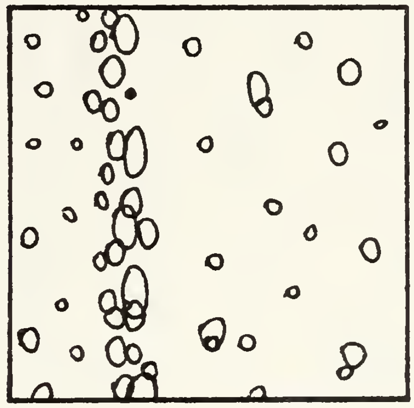
D. SUMMARY

The scenario section of the embedded tutorial is important to the development of qualified operators because it gives them the opportunity to refine their skills on tactical data and to experiment with different combinations of parameters. This is especially important in the early stages of development since there are no concrete guidelines for obtaining the best picture of the surveillance area.

type ==> Menu, Location, Peaks, Accuracy, Surface, Return

REGION NUMBER 6

COHERENCE LIMIT 1.3



GOOD SOLID PATTERN DOWN THE LEFT SIDE OF THE REGION.

FIGURE 6-14

EXCELLENT SURFACE-PLOT

type ==> Menu, Location, Peaks, Accuracy, Surface, Return

REGION NUMBER 1

COHERENCE LIMIT 1.0

NO PATTERN IN THIS LARGE AREA.

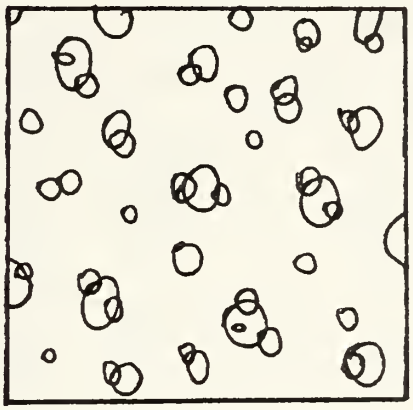
FIGURE 6-15

LARGE AREA SURFACE-PLOT

type ==> Menu, Location, Peaks, Accuracy, Surface, Return

REGION NUMBER 3

COHERENCE LIMIT 0.6



TOO MUCH NOISE DUE TO THE LOW COHERENCE LIMITS.

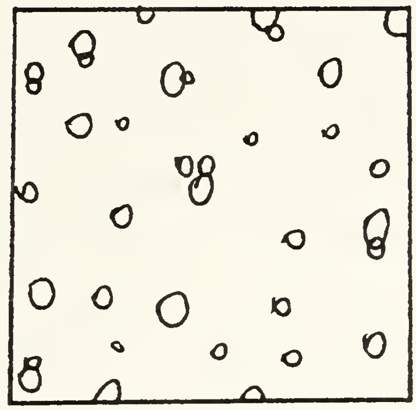
FIGURE 6-16

LOW COHERENCE SURFACE-PLOT

type ==> Menu, Location, Peaks, Accuracy, Surface, Return

REGION NUMBER 4

COHERENCE LIMIT 0.6



A LOT OF NOISE DUE TO TOO LOW COHERENCE LIMITS AND TOO LARGE AN AREA TO COVER ADEQUATELY.

FIGURE 6-17

WORST CASE SURFACE-PLOT

VII. CONCLUSION

The data obtained tends to indicate that computer assisted instruction is superior to conventional training when the trainee will eventually be working on a computer or console. This can be attributed to certain individuals being more inclined toward this type of work and thus the use of a computer based instruction seems quite natural. The failures in the use of computer based instruction tend to be in fields that require more manual dexterity, i.e. machinist mate and boiler technician, or where the trainee has a much lower initial aptitude. If the trainee has poor reading skills, this type of instruction will highlight all of his weaknesses, whereas some individuals can acquire a great deal of knowledge by listening to an instructor.

The disadvantage of computer based instruction in almost every case is the cost of the intial system compared to the cost of a teacher for the same number of students. An embedded tutorial reduces the cost since the computer is already paid for by its operational requirement. The only expense is possibly of an additional console and the cost of the instructional code. The extra console could be used as a spare for the operational system and the instructional code should not be too much more expensive than the printing of a textbook.

By the demonstration of the instuction presented in

this paper, it is hoped that future research can be done in the area of computer based instruction for the appropriate fields. The flexibility of the instruction, the ease of manipulation, and the variety of possible displays all point to the vast capabilities of a color graphics tutorial.

A. RECOMMENDATIONS

Further research should be conducted into the use of computer assisted instruction for certain types of jobs, i.e. ASW surveilance, secretarial positions, clerical positions, communications personnel. The natural abilities of personnel entering these fields makes the use of computers seem almost natural, since they generally have both a higher I.Q. and clerical aptitude.

This tutorial should be designed for the Ramtek 9400 since that is the equipment to be used operationally. The school is presently installing a Ramtek 9400 so work could continue on this project by transferring the code to the new system with improved performance because of the superior capabilities of the new console. The amount of code required by many of the displays could be reduced because the Ramtek 9400 can perform some functions easily which are quite complex on the existing system.

The use of color can enhance the pattern recognition capabilities of humans so a further project would be to display the surface plot data in different colors corresponding to signal amplitude.

Another project would be to write an algorithm that would take actual data from the operational system and transform it into displays that could be used for the scenario section of the tutorial. This would give the operator an almost limitless number of new examples on which to practice.

```
/*************************************************************
*                                                           *
* This is the main section of code that controls the other *
* separately compiled sections of code. It establishes the *
* type of tutorial that the user wishes to see and then it *
* forks to that section of code.                            *
*                                                           *
*************************************************************/
char select;    /* global variable */

main()  {
float x,y,z;
char cc,cd,ce,*st;
int i;
ramtek();  /*initializes the Ramtek system */
writon(1);
colortable();  /* establishes a new color table */
colort(10);
pic1();     /* allows the first display to remain */
cc='z';
while (cc!='s')  {   /* while #1 */
  cc=retchar();   }   /*end while #1 */
while (1)   {   /* while #2 */
 colort(10);
 pic99();   /* introduction display */
 cc='z';
 while(cc=='z')   {  /* while #3 */
  cc=retchar();
  switch(cc)  {   /* switch #1 */
    case 'a':   /* basic tutorial */
      select='a';
      break;
    case 'b':   /* data display tutorial */
      select='b';
      break;
    case 'c':   /* senario tutorial */
      select='c';
      break;
    case 'q':   /* Quit */
      pstop();
      pic99();   /* introduction display */
      cc='z';
      break;
    default:
      cc='z';
   }   /* end switch #1 */
 }   /* end while #3  */
color(15);
block(14.0,2.0,90.0,10.0);
strtxy(30.0,8.0);
color(0);
strout("Do you want any instructions? ( y / n )");
cc=retchar();
```

```
if (cc=='y')   {
  pic89();   /* instruction display */
  cc=retchar();
  while (cc!='c')
    cc=retchar();
 }
switch (select)   {   /* switch #2 */
  case 'a':   /* basic tutorial */
    colort(1);
    asw();   /* interrupt display */
    sleep(4);
    if((i=fork())==0)   {
      execl("seg1","seg1",0);
      exit();
     }
    wait(&i);
    break;
  case 'b':   /* data display */
    colort(2);
    asw();   /* interrupt display */
    sleep(4);
    if((i=fork())==0)   {
      execl("seg3","seg3",0);
      exit();
     }
    wait(&i);
    break;
  case 'c':   /* senario tutorial */
    colort(3);
    asw();   /* interrupt display */
    sleep(4);
    if((i=fork())==0)   {
      execl("Senerio","Senerio",0);
      exit();
     }
    wait(&i);
    break;
  default:
    printf("ERROR");
  }  /* end switch #2 */
 }   /* end while #2 */
}   /* end main */


colortable()  {   /* Establishes a new color table */
 int a[16];
     a[0] = triple(0,0,0);      /* black */
     a[1] = triple(15,0,0);     /* blue */
     a[2] = triple(0,15,0);     /* green */
     a[3] = triple(0,0,15);     /* red */
     a[4] = triple(15,5,0);     /* light blue */
     a[5] = triple(2,0,6);      /* maroon */
     a[6] = triple(0,10,0);     /* bright green */
```

```
     a[7] = triple(2,6,10);     /* peach */
     a[8] = triple(6,10,2);     /* pail green */
     a[9] = triple(10,2,6);     /* purple */
     a[10] = triple (13,5,2);   /* medium blue */
     a[11] = triple(5,2,13);    /* hot pink */
     a[12] = triple (2,13,5);   /* light green */
     a[13] = triple(15,0,15);   /* pink */
     a[14] = triple(10,4,4);    /* greyish blue */
     a[15] = triple(15,15,15);  /* white */
 clrtbl(10,a);
 return;
}

pstop()  {
     /* Displays the question of "Do you really want
        to quit?" and then waits for a reply       */
char *st,*sp,stop;
sp="   ";
color(11);
block(0.0,0.0,100.0,100.0);
color(5);
strtxy(45.0,80.0);
st="DO  YOU";
strout(st); strout(sp);
st="REALLY";
strout(st);
strout(sp);
st="WANT TO";
strout(st); strout(sp);
st=" QUIT?";
strout(st);
st="(y / n)";
strout(st);
stop=retchar();
if (stop=='y')  {
 color(0);
 block(0.0,0.0,100.0,100.0);
 exit();  /* Exits program if user want to quit */
 }
return;
}


pic1()  {  /* Displays ASW in large block letters */
 char *st1,*st2;
 color(3);
 block(0.0,0.0,100.0,100.0);
 color(12);
 block(15.0,50.0,20.0,68.0);
 block(30.0,50.0,35.0,68.0);
 block(15.0,58.0,35.0,63.0);
 block(15.0,68.0,21.0,69.0);
 block(16.0,69.0,34.0,70.0);
 block(17.0,70.0,33.0,71.0);
```

```
 block(18.0,70.0,32.0,72.0);
 block(19.0,71.0,31.0,73.0);
 block(29.0,68.0,34.0,69.0);
 block(40.0,50.0,60.0,55.0);
 block(55.0,50.0,60.0,63.0);
 block(40.0,58.0,60.0,63.0);
 block(40.0,58.0,45.0,73.0);
 block(40.0,68.0,60.0,73.0);
 block(65.0,50.0,70.0,73.0);
 block(80.0,50.0,85.0,73.0);
 block(74.0,53.0,76.0,60.0);
 block(73.0,52.0,74.0,59.0);
 block(72.0,51.0,73.0,58.0);
 block(71.0,50.0,72.0,57.0);
 block(70.0,50.0,71.0,56.0);
 block(76.0,52.0,77.0,59.0);
 block(77.0,51.0,78.0,58.0);
 block(78.0,50.0,79.0,57.0);
 block(79.0,50.0,80.0,56.0);
 dblwid(1);
 st1="EMBEDDED";
 st2="TUTORIAL";
 strtxy(40.0,40.0);
 strout(st1);
 strout(st2);
 dblwid(0);
 strtxy(70.0,5.0);
 color(15);
 strout("type s to start");
return;
}


pic99()   {   /* Introduction Display */
color(4);
block(0.0,0.0,100.0,100.0);
color(0);
dblwid(1);
strtxy(20.0,95.0);
strout("EMBEDDED TUTORIAL INTRODUCTION");
dblwid(0);
strtxy(15.0,87.0);
strout("    This tutorial is designed to meet the needs of an
          assortment of");
strout("users, from the novice to the experienced tracker.
         There are three");
strout("categories that can be reviewed:  ");
strout("   ");
strout("a) The basic principles of the system are discussed
           and many of the");
strout("    problems associated with the system's environment
           are dealt with");
strout("    in detail in this section. ");
strout("b) The types of data that will be displayed during
```

```
        actual system");
strout("   operation are presented along with explanations
           of the data and");
strout("   key areas to watch for while tracking. ");
strout("c) A set of practice senarios are presented based
           on the user inputted");
strout("   parameters.  These senarios allow the user to
           see actual data being");
strout("   presented and the effects of various parameter
           settings.  Key points");
strout("   are also highlighted in this section. ");
strtxy(15.0,8.0);
strout("type in the letter of the desired category (a,b,c)
         or (q) to quit");
return;
}

pic89()   {   /* Instruction Display */
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,30.0,100.0,90.0);
strtxy(10.0,97.0);
color(2);
strout("Menu of possible areas to proceed to next is
        presented here.");
strtxy(20.0,70.0);
switch(select)   {
  case 'a':
    strout("  Basic diagrams will be illustrated in this
           area. They will");
    strout("depict a view of the overall system, environ-
          mental problems,");
    strout("and detailed displays of questionable areas.
           Review this area");
    strout("before reading the explanation below. ");
    break;
  case 'b':
    strout("  This section will illustrate some typical
           data that would");
    strout("show up as the tracker is utilizing the
         operational system.");
    strout("The data will represent some relatively good
        tracks in order");
    strout("to demonstrate how the data should appear. ");
    break;
  case 'c':
    strout("  The data illustrated in this section is
        dependent upon the");
    strout("user inputs of parameters.  As the user alters
          the parameters");
    strout("on the menu display, the computer alters the
        type of data ");
    strout("that will be exhibited. ");
```

```
    break;
 }
strtxy(16.0,29.0);
switch(select)  {
  case 'a':
    strout("  An explanation of the above area will be
        displayed in this");
    strout("section.  Generally it will be a simplified
         review of the basic");
    strout("concept with a few new ideas.  If the new ideas
         are not fully");
    strout("understood, then they will probably be listed
          in the menu and");
    strout("can be pursued further. ");
    break;
  case 'b':
    strout("  An explanation of the above data and some key
         points");
  strout("to watch for will be described in this section.");
    break;
  case 'c':
    strout("  A very brief statement about the data and its
              reasons for");
    strout("being good or bad will be described in this
             section. ");
    break;
 }
strtxy(50.0,6.0);
strout("type c to continue");
return;
}


asw()   {   /* Interrupt Display */
float x;
int i;
i=6;
for(x=0.0;x<41.0;x=x+5.0)  {
  i=i+1;
  color(i);
  block(x,x,100.0-x,100.0-x);
 }
strtxy(45.0,58.0);
color(1);
strout(" A S W ");
strout("Embedded");
strout("Tutorial");
return;
}
```

```
/*#############################################################
#  This program demonstrates some of the capabilities of  #
#  a computer to instruct individuals in basic skills.    #
#  It is designed to be self-paced and totally contains   #
#  all the material required ( pictures, text, formulas,  #
#  and instructions). Once the user is logged on there is #
#  no need for an instructor to monitor the individual.   #
#  Due to the size of the program, it is broken up into   #
#  five modules that are compiled separately and then are #
#  forked together when required.  There are three parts  #
#  to the tutorial and then the senario section.  This is #
#  the first section of the tutorial.                     #
#############################################################*/


main()  {
float x,y,z;
char cc,cd,ce,*st;
int i;
ramtek();  /*initializes the Ramtek system */
writon(1);
colortable();  /* establishes a new color table */
colort(10);
pic1();     /* allows the first display to remain */
sleep(5);   /* on the screen 5 seconds before    */
pic2();     /* the second display is presented  */
cc='z';
while (cc=='z')  {   /* while #1 */
  cc=retchar();
  switch(cc)  {   /* switch #1 */
    case 'c':   /* Continue */
     pic3();
     cd='z';
     while (cd=='z')  {    /* while #2 */
       cd=retchar();
       switch(cd)  {   /* switch #2 */
         case 'c':   /* Continue */
           break;
         case 'r':   /* Return */
           pstop();
           pic3(); /* displays pic3 if quit isn't desired */
           cd='z';
           break;
         default:
          cd='z';
       }   /*end switch #2 */
     }  /* end while #2 */
    break;
   case 'f': /* Extra display if 'f' is accidently typed*/
     pic2a();
     cc='z';
```

```
        break;
      case 'r':   /* Return */
        pstop();
        pic3();  /* displays pic 3 if quit isn't desired */
        cc='z';
        break;
      default:
        cc='z';
    }   /* end switch #1 */
}   /* end while #1 */
pic4();
cc='z';
while (cc=='z')  {   /* while #3 */
  cc=retchar();
  switch(cc)  {   /* swithch #3 */
    case 'n':   /* Noise display */
      pic31();
      cc='z';
      break;
    case 'd':   /* Distance display */
     pic32();
     cd='z';
     while (cd=='z')  {   /* while #4 */
       cd=retchar();
       switch(cd)  {   /* switch #4 */
         case 'n':   /* Noise display */
           pic31();
           cc='z';   /* steps out of while #4 */
           break;
         case 't':   /* Time display */
           pic33();
           cc='z';   /* steps out of while #4 */
           break;
         case 'w':   /* Water temperature display */
           pic34();
           cc='z';   /* steps out of while #4 */
           break;
         case 'l':   /* Tau line display */
           pic321();
           ce='z';
           while (ce=='z')  {   /* while #5 */
             ce=retchar();
             switch(ce)  {   /* switch #5 */
               case 'n':   /* Noise display */
                 pic31();
                 cc='z';  /* steps out of while #4 */
                 break;
               case 't':   /* Time display */
                 pic33();
                 cc='z';  /* steps out of while #4 */
                 break;
               case 'd':   /* Distance display */
                 pic32();
                 cc='z';  /* steps out of while #4 */
```

```
            break;
          case 'w':   /* Water temperature display */
            pic34();
            cc='z';  /* steps out of while #4 */
            break;
          case 'p':  /* Tau line problems */
            pic3211();
            cc='z';  /* steps out of while #4 */
            break;
          case 'c':   /* Continue */
            pic5();  /* steps out of while #3 */
            break;
          case 'r':   /* Return */
            pstop();
            pic321();  /* displays pic 321 if quit */
            ce='z';    /* isn't desired           */
            break;
          default:
            ce='z';
        }   /* end switch #5 */
      }   /* end while #5 */
      break;
    case 'c':   /* Continue */
      pic5();   /* steps out of while #3 */
      break;
    case 'r':   /* Return */
      pstop();
      pic32();/*displays pic 32 if quit isn't desired*/
      cd='z';
      break;
    default:
      cd='z';
  }   /* end switch #4 */
 }   /* end while #4 */
 break;
case 't':   /* Time display */
  pic33();
  cc='z';
  break;
case 'w':  /* Water temperature display */
  pic34();
  cc='z';
  break;
case 'c':   /* Continue */
  pic5();
  break;
case 'r':   /* Return */
  pstop();
  pic4();   /* displays pic 4 if quit isn't desired */
  cc='z';
  break;
default:
  cc='z';
}   /* end switch #3 */
```

```
}   /* end while #3 */
if((i=fork())==0)  {    /* If stmt #1 */
  execl("seg2","seg2",0);    /* forks to segment 2 */
  exit();
  }   /* end If stmt #1 */
wait(&i);
}   /* end of main */



colortable()  {    /* Establishes a new color table */
 int a[16];
     a[0] = triple(0,0,0);       /* black */
     a[1] = triple(15,0,0);      /* blue */
     a[2] = triple(0,15,0);      /* green */
     a[3] = triple(0,0,15);      /* red */
     a[4] = triple(15,5,0);      /* light blue */
     a[5] = triple(2,0,6);       /* maroon */
     a[6] = triple(0,10,0);      /* bright green */
     a[7] = triple(2,6,10);      /* peach */
     a[8] = triple(6,10,2);      /* pail green */
     a[9] = triple(10,2,6);      /* purple */
     a[10] = triple (13,5,2);    /* medium blue */
     a[11] = triple(5,2,13);     /* hot pink */
     a[12] = triple (2,13,5);    /* light green */
     a[13] = triple(15,0,15);    /* pink */
     a[14] = triple(10,4,4);     /* greyish blue */
     a[15] = triple(15,15,15);   /* white */
 clrtbl(10,a);
 return;
}



pstop()  {
           /* Displays the question of "do you really want
              to quit?" and then sets the value of STOP  */
char *st,*sp;
sp="  ";
color(11);
block(0.0,0.0,100.0,100.0);
color(5);
strtxy(45.0,80.0);
st="DO  YOU";
strout(st); strout(sp);
st="REALLY";
strout(st);
strout(sp);
st="WANT TO";
strout(st); strout(sp);
st=" RETURN TO";
strout(sp);
strout("THE MAIN"); strout(sp);
strout("PROGRAM?");
```

```
strout(st);
st="(y / n)";
strout(st);
STOP=retchar();
if (STOP=='y')  {
 STOP='Y';
 color(0);
 block(0.0,0.0,100.0,100.0);
 }
return;
}



pic1()  {   /* Displays ASW in large block letters */
 char *st1,*st2;
 color(3);
 block(0.0,0.0,100.0,100.0);
 color(12);
 block(15.0,50.0,20.0,68.0);
 block(30.0,50.0,35.0,68.0);
 block(15.0,58.0,35.0,63.0);
 block(15.0,68.0,21.0,69.0);
 block(16.0,69.0,34.0,70.0);
 block(17.0,70.0,33.0,71.0);
 block(18.0,70.0,32.0,72.0);
 block(19.0,71.0,31.0,73.0);
 block(29.0,68.0,34.0,69.0);
 block(40.0,50.0,60.0,55.0);
 block(55.0,50.0,60.0,63.0);
 block(40.0,58.0,60.0,63.0);
 block(40.0,58.0,45.0,73.0);
 block(40.0,68.0,60.0,73.0);
 block(65.0,50.0,70.0,73.0);
 block(80.0,50.0,85.0,73.0);
 block(74.0,53.0,76.0,60.0);
 block(73.0,52.0,74.0,59.0);
 block(72.0,51.0,73.0,58.0);
 block(71.0,50.0,72.0,57.0);
 block(70.0,50.0,71.0,56.0);
 block(76.0,52.0,77.0,59.0);
 block(77.0,51.0,78.0,58.0);
 block(78.0,50.0,79.0,57.0);
 block(79.0,50.0,80.0,56.0);
 dblwid(1);
 st1="EMBEDDED";
 st2="TUTORIAL";
 strtxy(40.0,40.0);
 strout(st1);
 strout(st2);
 dblwid(0);
return;
}
```

```
pic2()  {     /* This subroutine displays the instructions
                 needed to use the tutorial  */
char *st;
int stime;
color(1);
block(0.0,0.0,100.0,100.0);
color(2);
dblwid(1);
strtxy(25.0,96.0);
st="TUTORIAL INSTRUCTIONS";
strout(st);
vector(25.0,92.0,72.0,92.0);
dblwid(0);
strtxy(10.0,91.0);
st="  This embedded tutorial is designed to give you the
     hands on training";
strout(st);
st="required to operate this ASW surveilance system.  It
    contains some overall";
strout(st);
st="views of the system, some theory of its working capabil-
    ities and then a short";
strout(st);
st="scenerio of possible actions which can be taken by you";
strout(st);
st="   ";
strout(st);
st="  To operate the tutorial there are three areas to watch
strout(st);
st="1) all of the instructions at the top of each frame
     provide";
strout(st);
st="   you with the possible areas you can proceed to next";
strout(st);
st="2) The middle area is either a display which will be";
strout(st);
st="normally viewed while operating the system or it is a";
strout(st);
st="display that provides a better view of the entire system
strout(st);
st="3) The bottom area gives you an explanation of the
      middle area.";
strout(st);
st="   ";
strout(st);
st="To utilize the tutorial, press the key of the first
     letter";
strout(st);
st="of the word which describes where you wish to proceed.";
strout(st);
st="    type ==> Continue, Filter, Quit";
strout(st);
st="By typing an f you would proceed to a display of filters
```

```
strout(st);
st="                NOW TYPE  c  TO CONTINUE";
strout(st);
return;
}



pic2a()  {
           /* This subroutine is displayed if a mistake is
              committed by the user on the instruction
              display  */
char *st;
color(2);
block(0.0,0.0,100.0,100.0);
color(12);
strtxy(50.0,50.0);
st="TYPE c";
strout(st);
return;
}


pic3()  {
       /* This subroutine calls for the display to be set
          up and tells about the purpose of the system  */
char *st;
color(1);
block(0.0,0.0,100.0,100.0);
ocean();
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Quit";
strout(st);
strtxy(4.0,28.5);
st="  This system is designed to pick up frequency changes
    in the ocean and determine";
strout(st);
st="whether or not they were made by a submarine or other
    vessel.  Assuming there";
strout(st);
st="was an area(green square) in the ocean where you wished
    to watch for submarines,";
strout(st);
st="all that would be necessary are the two arrays of
    hydrophones (1 and 2) and";
strout(st);
st="the technique to analyze their outputs.  For better
     results additional arrays can";
strout(st);
st="be added, for instance 3.";
strout(st);
return;
}
```

```
ocean()  {
        /* This subroutine draws the basic display of the
           ocean, area under surveilance, land, and the
           arrays   */
char *st;
color(4);
block(0.0,29.0,100.0,90.0);
land();
arrays();
color(6);
block(10.0,60.0,26.0,80.0);
return;
}


arrays()  {   /* Draws the arrays */
color(7);
block(79.0,84.0,80.0,79.0);
block(25.0,32.0,29.0,33.0);
vector(79.0,33.9,82.0,37.1);
vector(79.5,33.8,82.5,37.2);
vector(79.7,33.7,82.7,37.3);
vector(80.0,33.6,83.0,37.4);
vector(79.2,33.5,82.2,37.5);
vector(79.3,33.4,82.3,37.6);
return;
}

land()  {   /* Draws the land  */
color(5);
block(84.0,50.0,100.0,90.0);
block(95.0,30.0,100.0,55.0);
block(89.0,40.0,96.0,51.0);
block(92.0,33.0,96.0,41.0);
vector(89.0,40.0,84.0,50.0);
vector(89.0,40.0,84.2,50.0);
vector(89.0,40.0,84.4,50.0);
vector(89.0,40.0,84.6,50.0);
vector(84.0,50.0,82.0,63.0);
vector(84.0,50.0,82.2,63.0);
vector(84.0,50.0,82.4,63.0);
vector(84.0,50.0,82.6,63.0);
vector(82.0,67.0,83.0,73.5);
vector(82.1,67.0,83.0,73.5);
vector(82.2,67.0,83.0,73.5);
vector(82.3,67.0,83.0,73.5);
vector(83.0,84.0,81.0,90.0);
vector(83.0,84.0,81.2,90.0);
vector(83.0,84.0,81.4,90.0);
vector(83.0,84.0,81.6,90.0);
vector(97.0,29.0,95.0,30.0);
vector(95.0,30.0,92.0,33.0);
```

```
vector(92.0,33.0,89.0,40.0);
vector(92.0,33.0,89.2,40.0);
vector(92.0,33.0,89.4,40.0);
vector(92.0,33.0,89.6,40.0);
block(82.8,78.0,83.0,80.0);
block(88.5,40.0,90.0,43.0);
block(82.8,57.0,84.0,60.0);
vector(92.0,33.0,89.8,40.0);
block(97.0,29.0,100.0,31.0);
block(86.0,45.0,90.0,51.0);
block(83.0,58.0,86.0,90.0);
block(96.0,29.0,98.0,31.0);
block(95.5,29.5,98.0,31.0);
block(96.5,29.0,98.0,30.0);
vector(97.1,29.0,95.0,30.2);
block(93.5,31.2,96.0,34.0);
block(92.7,32.0,98.0,34.0);
block(90.0,36.6,94.0,41.0);
block(91.0,34.4,94.0,41.0);
block(87.2,42.2,90.0,46.0);
block(84.8,47.2,88.0,51.0);
block(82.0,62.7,85.0,67.0);
block(82.0,88.0,85.0,90.0);
block(82.5,66.0,84.0,71.0);
block(82.5,86.0,84.0,89.0);
block(83.5,53.0,85.0,60.0);
block(82.5,60.0,84.0,65.0);
return;
}



pic31()  {   /* Noise display */
char *st;
color(1);
block(0.0,0.0,100.0,100.0);
color(7);
block(0.0,29.0,100.0,90.0);
color(2);
st="type ==> Continue, Distance, Time, Water(temp), Quit";
strtxy(10.0,98.0);
strout(st):
strtxy(4.0,27.5);
st=" Noise is what we are attempting to detect in the ocean,
     however there is other";
strout(st);
st="noise in the water. This detection can be compared to
     trying to listen to a";
strout(st);
st="specific person at a large noisy party. The majority of
    noise in the range";
strout(st);
st="of 10Hz to 400Hz is caused by other vessels in the water,
    probably at a much";
```

```
strout(st);
st="larger distance and thus a much weaker signal is
    received.";
strout(st);
color(0);
vector(30.0,40.0,30.0,80.0);
vector(30.0,40.0,75.0,40.0);
vector(30.0,45.0,75.0,45.0);
block(50.0,45.0,50.5,62.0);
return;
}


pic32()  {
              /* This display discusses the problems of
                 large distances  */
char *st;
color(1);
block(0.0,90.0,100.0,100.0);
block(0.0,0.0,100.0,28.0);
ocean();
color(7);
block(15.0,65.0,18.0,68.0);
color(0);
vector(17.0,66.5,27.0,33.0);
vector(17.0,66.5,80.0,35.0);
strtxy(24.5,48.0);
st="500 miles";
strout(st);
strtxy(50.0,55.0);
st="1200 miles";
strout(st);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Noise, Time, Water(temp), Lines(tau)
      , Quit";
strout(st);
strtxy(4.0,28.5);
st="  Due to the mission of this system it is necessary to
    detect any vessel at a";
strout(st);
st="very large distance.  These distances can range up over
   1200 miles or they can";
strout(st);
st="be relatively short.  The long ranges have problems of
    diffusion and large";
strout(st);
st="separations in their tau lines.  While using two arrays
    with large distance";
strout(st);
st="differences causes problems in time of arrival of the
    signal at the various arrays.";
strout(st);
return;  }
```

```
pic321() {  /* This subroutine discusses tau lines */
char *st;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(6);
vector(72.0,68.0,72.0,34.9);
vector(37.6,51.4,72.0,51.4);
vector(37.6,75.8,49.0,68.8);
vector(49.0,68.8,61.4,62.9);
vector(61.4,62.9,72.0,59.8);
color(8);
strtxy(71.0,69.5);
st="-- A";
strout(st);
strtxy(73.0,61.5);
st="3 => 500 tau line";
strout(st);
strtxy(73.0,53.5);
st="1 => 0 tau line";
strout(st);
strtxy(71.0,36.5);
st="-- B";
strout(st);
st="2";
strtxy(36.0,53.5);
strout(st);
strtxy(36.0,78.0);
st="4";
strout(st);
strtxy(10.0,98.0);
color(2);
st="type ==> Continue, Noise, Distance, Time, Water(temp),
    Problems(tau line), Quit";
strout(st);
strtxy(4.0,28.5);
st="  Tau lines are just measurement lines distinguishing
    the time arrival of the";
strout(st);
st="signal at two different arrays.  If arrays A and B are
    1000 miles apart then a";
strout(st);
st="signal produced exactly between A and B (point 1) would
    arrive at both arrays at";
strout(st);
st="exactly the same time. Additionally any signal produced
    on line 1-2 would arrive";
strout(st);
st="at both arrays simultaneously. In the same manner, any
    signal produced on line 3-4";
strout(st);
```

```
st="arrive at the arrays with the same amount of time delay
   in all cases.";
strout(st);
return;
}

pic3211() {
      /* This subroutine discusses the problems associated
         with the tau lines and large distances  */
char *st;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(6):
vector(72.0,68.0,72.0,34.9);
vector(37.6,51.4,72.0,51.4);
vector(37.6,75.8,49.0,68.8);
vector(49.0,68.8,61.4,62.9);
vector(61.4,62.9,72.0,59.8);
vector(37.6,53.7,72.0,53.1);
vector(37.6,83.3,49.0,74.0);
vector(49.0,74.0,60.8,65.8);
vector(60.8,65.8,72.0,61.7);
color(8);
strtxy(71.0,69.5);
st=" -- A";
strout(st);
strtxy(73.0,61.5);
st="3 => 500 tau line";
strout(st);
strtxy(73.0,53.5);
st="1 => 0 tau line";
strout(st);
strtxy(71.0,36.5);
st=" -- B";
strout(st);
st="2";
strtxy(36.0,53.5);
strout(st);
strtxy(36.0,78.0);
st="4";
strout(st);
st="225 miles --->";
strtxy(20.0,81.0);
strout(st);
st="65 miles --->";
strtxy(20.0,54.0);
strout(st);
strtxy(10.0,98.0);
color(2);
st="type ==> Continue, Noise, Distance, Time, Water(temp)";
strout(st);
strtxy(4.0,28.5);
```

```
st="  Since the tau lines are hyperbolic in shape, another
    problem arises in that the";
strout(st);
st="spacing between any two lines does not remain constant.
    The lines at points 1 and 3";
strout(st);
st="are each spaced at 50 miles apart. The lines 1-2 spread
    apart to 65 miles by the";
strout(st);
st="time they reach point 2, whereas the lines 3-4 spread
   apart to 225 miles by the time";
strout(st);
st="they reach point 4.  This variance can cause drastic
    differences in the accuracy";
strout(st);
st="of the data received, depending on it's position
    relative to the arrays.";
strout(st);
return;
}

pic33()  {
          /* This subroutine discusses the problems of time
             delay and how the computer solves them  */
char *st;
color(1);
block(0.0,90.0,100.0,100.0);
block(0.0,0.0,100.0,28.0);
ocean();
color(0);
vector(17.0,66.5,27.0,33.0);
vector(17.0,66.5,80.0,35.0);
strtxy(16.0,70.0);
st="A";
strout(st);
strtxy(24.5,48.0);
st="500 miles";
strout(st);
strtxy(50.0,55.0);
st="1200 miles";
strout(st);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Noise, Distance, Water(temp), Quit";
strout(st);
strtxy(4.0,28.5);
st="  Since sound travels at 2989 Kts. (3400 miles/hour)
    through the water you";
strout(st);
st="can approximate it as travelling at 1 mile per second.
   At this speed it would take";
strout(st);
st="a noise particle 1200 sec. or 20 minutes to travel from
    point A to array 2. But";
```

```
strout(st);
st="it would only take 8 minutes for it to travel to array
    3, thus arriving 12 minutes";
strout(st);
st="before it would arrive at array 2. This difference must
    be accounted for by";
strout(st);
st="the algorithum in the computer.";
strout(st);
return;
}

pic34() {
          /* This subroutine demonstrates how the speed of
             sound travels at different speed depending on
             temperature, salinity, and depth   */
char *st;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(6);
vector(41.0,76.0,63.0,76.0);
vector(41.0,76.0,41.0,36.0);
strtxy(34.0,77.8);
st="   0 --";
strout(st);
st="1000 --";
strtxy(34.0,70.0);
strout(st);
st="2000 --";
strtxy(34.0,62.0);
strout(st);
st="3000 --";
strtxy(34.0,54.0);
strout(st);
st="4000 --";
strtxy(34.0,46.0);
strout(st);
st="5000 --";
strtxy(34.0,38.0);
strout(st);
strtxy(39.0,81.0);
st="1475      1500       1525";
strout(st);
st="  1         1          1";
strtxy(38.5,78.0);
strout(st);
st="Velocity (yards/sec)";
strtxy(40.0,87.0);
strout(st);
st=" Depth";
strtxy(24.0,66.0);
strout(st);
```

```
st="(yards)";
strout(st);
color(8);
vector(53.0,76.0,54.4,74.2);
vector(54.4,74.2,54.4,73.0);
vector(54.4,73.0,46.6,70.2);
vector(46.6,70.2,44.2,68.1);
vector(44.2,68.1,44.0,66.0);
vector(44.0,66.0,44.3,64.0);
vector(44.3,64.0,45.8,60.0);
vector(45.8,60.0,58.0,36.0);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Noise, Time, Distance, Quit";
strout(st);
strtxy(4.0,28.5);
st="The velocity of sound changes as the water temperature,
    salinity, and depth vary.";
strout(st);
st="In the first 300 yards from the surface velocity is
    mainly altered by temperature and";
strout(st);
st="salinity, after which depth (pressure) is the main
    factor.  As temperature and salinity";
strout(st);
st="rise the velocity of sound increases.  As the pressure
    increases due to the depth,";
strout(st);
st="the velocity of sound also increases.  These variances
    are not severe but when the";
strout(st);
st="measurements are in tenths of seconds, any change will
    cause inaccuracies.";
strout(st);
return;
}

pic4()  {
        /* This subroutine shows in general what all the
           problems are in the environment that must be
           overcome by the system   */
char *st;
color(1);
block(0.0,90.0,100.0,100.0);
block(0.0,0.0,100.0,28.0);
ocean();
color(7);
block(15.0,65.0,18.0,68.0);
color(0);
vector(17.0,66.5,27.0,33.0);
vector(17.0,66.5,80.0,35.0);
strtxy(24.5,48.0);
st="500 miles";
strout(st);
```

```
st="1200 miles";
strtxy(50.0,55.0);
strout(st);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Noise, Distance, Time, Water(temp)";
strout(st);
strtxy(4.0,28.5);
st="  There are several problems in the system that are
    difficult to overcome:";
strout(st);
st="1) The distances are quite large, on the order of 1200";
strout(st);
st="2) Large differences in distances from area to arrays
    cause significant time delays.";
strout(st);
st="3) Differences in water temperature and salinity cause
     the sounds to travel at";
strout(st);
st="   various speeds.";
strout(st);
st="4) Other noises are always present in the ocean.";
strout(st);
return;
}

pic5()  {
           /* This subroutine shows how the large area is
              broken up into small regions   */
char *st;
color(1);
block(0.0,90.0,100.0,100.0);
block(0.0,0.0,100.0,28.0);
ocean();
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Quit";
strout(st);
strtxy(4.0,28.5);
st="  These large distances between arrays allow the system
     to cover";
strout(st);
st="a large area of ocean, let's say 400 miles by 400 miles
   or roughly";
strout (st);
st="the size of Montana. This area can be further subdivided
    into 100 sectors,";
strout(st);
st="each 40 miles square to make the tracking easier.  Each
    of these sectors";
strout(st);
st="can presently be covered by XXX correlations per hour.";
strout(st);
disect();
```

```
color(13);
block(19.6,64.3,21.0,66.0);
color(0);
st="400 miles";
strtxy(14.0,84.0);
strout(st);
strtxy(27.0,72.0);
strout(st);
return;
}

disect()  {  /* This subroutine disects the region */
float x1,x2,y1,y2;
color(0);
x2=26.0;
y1=60.0;
y2=80.0;
for (x1=10.0;x1<=27.0;x1=x1+1.6)
 vector(x1,y1,x1,y2);
x1=10.0;
for (y1=60.0;y1<=81.0;y1=y1+2.0)
 vector(x1,y1,x2,y1);
return;
}   /* End of segment 1  */
```

```
/*###############################################################
#  This is the third  section of the tutorial and it is     #
# set up in the same manner as the other sections. It con-  #
# tinues the presentation of displays and allows the user   #
# to see the overall system and some of its fine points.    #
# Many of the parts that were generated in the first part   #
# have to be redone in this section due to the forking      #
# technique used to link compiled programs.                 #
###############################################################*/


main()  {   /* section three of the tutorial */
float x,y,z;
char cc,cd,ce,*st;
int j,w;
ramtek();  /* initializes the Ramtek system */
writon(1);
colortable();  /* establishes a color table */
colort(10);
pic5();   /* area breakdown display */
w=0;
cc='z';
while (cc=='z')  {   /* while #1 */
  cc=retchar();
  switch(cc)  {    /* switch #1 */
    case 'c':   /* Continue */
      pic6();  /* correlation display */
      break;
    case 'r':   /* Return */
      pstop();
      pic6();   /* correlation display */
      cc='z';
      break;
    default:
      cc='z';
   }   /* end switch #1 */
 }   /* end while #1 */
cc='z';
while (cc=='z')  {    /* while #2 */
  cc=retchar();
  switch(cc)  {     /* switch #2 */
    case 'h':   /* hydrophone display */
      pic61();
      cd='z';
      while (cd=='z')  {   /* while #3 */
        cd=retchar();
        switch(cd)  {    /* switch #3 */
          case 'a':   /* array pointing display */
            pic611();
            cc='z';
            break;
          case 'e':   /* ellipse display */
```

```
          pic62();
          ce='z';
          while (ce=='z')  {   /* while #4 */
            ce=retchar();
            switch(ce)  {    /* switch #4 */
              case 'c':    /* Continue */
                if (w==1)
                   /* If combined ellipses display
                      has been presented, skip and
                      display pic8      */
                  pic8();
                else
                  pic7();
                break;
              case 'e':   /* combined ellipses display */
                pic621();
                w=1;
                cc='z';
                break;
              case 'h':   /* hydrophone display */
                pic61();
                cd='z';
                break;
              case 'r':   /* Return */
                pstop();
                pic61();  /* hydrophone display */
                cd='z';
                break;
              default:
                ce='z';
              }   /* end switch #4 */
            }   /* end while #4 */
          cc='z';
          break;
        case 'c':   /* Continue */
          pic7();
          break;
        case 'r':  /* Return */
          pstop();
          pic61();   /* hydrophone display */
          cd='z';
          break;
        default:
          cd='z';
      }   /* end switch #3 */
    }   /* end while #3 */
  break;
case 'e':   /* ellipse display */
  pic62();
  cd='z';
  while (cd=='z')  {   /* while #5 */
    cd=retchar();
    switch(cd)  {    /* switch #5 */
      case 'e':   /* combined elipses display */
```

```
          pic621();
          w=1;
          cc='z';
          break;
        case 'h':   /* hydrophone display */
          pic61();
          ce='z';
          while(ce=='z') {  /* while #6 */
            ce=retchar();
            switch(ce)  {  /* switch #6 */
              case 'c':    /* Continue */
                if (w==1)
                  pic8();
                else
                  pic7();
                break;
              case 'a':   /*  array pointing display */
                pic611();
                cc='z';
                break;
              case 'e':   /* ellipse display */
                pic62();
                cd='z';
                break;
              case 'r':   /* Return */
                pstop();
                pic62();   /* ellipse display */
                cd='z';
                break;
              default:
                ce='z';
             }    /* end of switch #6 */
           }    /* end of while #6 */
          break;
        case 'c':   /* Continue */
          if (w==1)
            pic8();
          else
            pic7();
          break;
        case 'r':  /* Return */
          pstop();
          pic62();   /* ellipse display */
          cd='z';
          break;
        default:
          cd='z';
      }   /* end switch #5 */
    }   /* end while #5 */
    break;
  case 'c':   /* Continue */
    if (w==1)
      pic8();
    else
```

```
        pic7();
      break;
    case 'r':   /* Return */
      pstop();
      pic6();   /* correlation display */
      cc='z';
      break;
    default:
      cc='z';
    }   /* end switch #2 */
  }   /* end while #2 */
 if (w==0)  {    /* If stmt #1 */
   cc='z';
   while (cc=='z')  {    /* while #7 */
     cc=retchar();
     switch(cc)  {   /* switch #7 */
       case 'm':   /* draws more ellipses */
         pic7a();
         color(1);
         block(0.0,90.0,100.0,100.0);
         color(2);
         strtxy(10.0,98.0);
         st="type ==> Continue, Return";
         strout(st);
         cc='z';
         break;
       case 'c':    /* Continue */
         pic8();
         break;
       case 'r':   /* Return */
         pstop();
         pic7();    /* combined ellipses display */
         cc='z';
         break;
       default:
         cc='z';
       }    /* end switch #7 */
     }    /* end while #7 */
  }    /* end If stmt #1 */
 else
  w=0;
 return;
}     /* end main section of segment 2 */


pstop()  {
char *st,*sp,stop;
   sp="  ";
   color(11);
   block(0.0,0.0,100.0,100.0);
   color(5);
   strtxy(45.0,80.0);
   st="DO  YOU";
   strout(st); strout(sp);
```

```
   st="REALLY";
   strout(st); strout(sp);
   st="WANT TO";
   strout(st); strout(sp);
   st=" RETURN TO";
   strout("THE MAIN"); strout(sp);
   strout("PROGRAM?");
   st="(y / n)";
   strout(st);
   stop=retchar();
   if (stop=='y')  {
     color(0);
     block(0.0,0.0,100.0,100.0);
     exit();
    }
return;
}



pic61()  {     /* Hydrophone display */
char *st;
float x1;
color(1);
block(0.0,0.0,100.0,100.0);
color(7);
block(0.0,29.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Ellipses, Array (pointing), Quit";
strout(st);
strtxy(4.0,27.0);
st="  The system for collecting noise data is an array of
    hydrophones, roughly 2000 feet";
strout(st);
st="long containing 60 passive sonar devices.  If no
    adjusting (pointing) of the array";
strout(st);
st="is done, then only noise waves coming in at the array
    directly abeam of the array";
strout(st);
st="will be detected.  The noise arrives at the array in
    the same manner that a wave";
strout(st);
st="crashes on the beach, some come in straight and others
    hit at an angle";
strout(st);
color(12);
block(19.0,37.0,81.0,39.0);
color(0);
for(x1=20.0;x1<80.5;x1=x1+1.0)
  block(x1,37.9,x1+0.2,38.1);
strtxy(15.0,80.0);
strout("Noise");
```

```
strout("Wave");
strtxy(54.0,70.0);
strout("Beam Pattern");
vector(22.0,76.0,78.0,76.0);
vector(22.0,52.0,78.0,52.0);
strtxy(42.0,35.5);
strout("HYDROPHONE ARRAY");
arrows();
color(13);
vector(50.0,39.0,48.0,72.0);
vector(50.0,39.0,52.0,72.0);
vector(48.0,72.0,48.2,77.0);
vector(48.2,77.0,49.0,79.0);
vector(49.0,79.0,51.0,79.0);
vector(51.0,79.0,51.8,77.0);
vector(51.8,77.0,52.0,72.0);
return;
}

arrows()  {
color(0);
strtxy(29.0,80.0);
strout("V       V                     V          V");
strtxy(29.0,56.0);
strout("V       V                     V          V");
vector(29.5,84.0,29.5,77.0);
vector(38.2,84.0,38.2,77.0);
vector(59.0,84.0,59.0,77.0);
vector(69.9,84.0,69.9,77.0);
vector(29.5,60.0,29.5,53.0);
vector(38.2,60.0,38.2,53.0);
vector(59.0,60.0,59.0,53.0);
vector(69.9,60.0,69.9,53.0);
return;
}

pic611()  {
char *st;
float x1;
color(1);
block(0.0,0.0,100.0,100.0);
color(7);
block(0.0,29.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Hydrophones, Ellipses, Quit";
strout(st);
strtxy(4.0,27.0);
strout("  In order to cover a large area of water with a
     single array, it must be");
strout("capable of directing its beam in several directions.
      This is accomplished");
strout("by using a time delay between receiving a signal at
      various hydrophones.");
```

```
strout("There can be roughly as many beam patterns as there
       are hydrophones and");
strout("the beam patterns generally overlap so as to cover
       the entire area.");
color(12);
block(39.0,37.0,61.0,39.0);
color(0);
for(x1=40.0;x1<60.5;x1=x1+1.0)
  block(x1,37.9,x1+0.2,38.1);
vector(50.0,39.0,48.0,72.0);
vector(50.0,39.0,52.0,72.0);
vector(48.0,72.0,48.2,77.0);
vector(48.2,77.0,49.0,79.0);
vector(49.0,79.0,50.0,79.3);
vector(50.0,79.3,51.0,79.0);
vector(51.0,79.0,51.8,77.0);
vector(51.8,77.0,52.0,72.0);
         /* first beam left */
vector(50.0,39.0,41.7,71.2);
vector(41.7,71.2,41.0,75.0);
vector(41.0,75.0,41.8,77.5);
vector(41.8,77.5,43.2,77.9);
vector(43.2,77.9,44.6,76.2);
vector(44.6,76.2,45.7,72.4);
vector(45.7,72.4,50.0,39.0);
        /* second beam left */
color(2);
vector(50.0,39.0,44.8,72.2);
vector(44.8,72.2,44.2,76.0);
vector(44.2,76.0,45.5,78.1);
vector(45.5,78.1,46.0,78.4);
vector(46.0,78.4,47.0,78.2);
vector(47.0,78.2,48.2,76.0);
vector(48.2,76.0,48.7,72.5);
vector(48.7,72.5,50.0,39.0);
        /* second beam right */
vector(50.0,39.0,51.5,72.5);
vector(51.5,72.5,52.0,76.2);
vector(52.0,76.2,53.2,78.4);
vector(53.2,78.4,55.5,78.0);
vector(55.5,78.0,56.2,75.0);
vector(56.2,75.0,55.8,72.3);
vector(55.8,72.3,50.0,39.0);
        /* first beam right */
color(0);
vector(50.0,39.0,54.8,72.2);
vector(54.8,72.2,56.0,76.0);
vector(56.0,76.0,57.4,77.9);
vector(57.4,77.9,59.2,77.3);
vector(59.2,77.3,59.6,75.0);
vector(59.6,75.0,58.7,71.5);
vector(58.7,71.5,50.0,39.0);
return;
}
```

```
colortable()  {  /* Establishes a new color table  */
 int a[16];
 a[0] = triple(0,0,0);
 a[1] = triple(15,0,0);
 a[2] = triple(0,15,0);
 a[3] = triple(0,0,15);
 a[4] = triple(15,5,0);
 a[5] = triple(2,0,6);
 a[6] = triple(0,10,0);
 a[7] = triple(2,6,10);
 a[8] = triple(6,10,2);
 a[9] = triple(10,2,6);
 a[10] = triple (13,5,2);
 a[11] = triple(5,2,13);
 a[12] = triple (2,13,5);
 a[13] = triple(15,0,15);
 a[14] = triple(10,4,4);
 a[15] = triple(15,15,15);
 clrtbl(10,a);
 return;
}

ocean()  {   /* Draws the ocean  */
char *st;
color(4);
block(0.0,29.0,100.0,90.0);
land();
arrays();
color(6);
block(10.0,60.0,26.0,80.0);
return;
}


arrays()  {  /* Draws the arrays  */
color(7);
block(79.0,84.0,80.0,79.0);
block(25.0,32.0,29.0,33.0);
vector(79.0,33.9,82.0,37.1);
vector(79.5,33.8,82.5,37.2);
vector(79.7,33.7,82.7,37.3);
vector(80.0,33.6,83.0,37.4);
vector(79.2,33.5,82.2,37.5);
vector(79.3,33.4,82.3,37.6);
return;
}


land()  {   /* Draws the land   */
color(5);
block(84.0,50.0,100.0,90.0);
block(95.0,30.0,100.0,55.0);
```

```
block(89.0,40.0,96.0,51.0);
block(92.0,33.0,96.0,41.0);
vector(89.0,40.0,84.0,50.0);
vector(89.0,40.0,84.2,50.0);
vector(89.0,40.0,84.4,50.0);
vector(89.0,40.0,84.6,50.0);
vector(84.0,50.0,82.0,63.0);
vector(84.0,50.0,82.2,63.0);
vector(84.0,50.0,82.4,63.0);
vector(84.0,50.0,82.6,63.0);
vector(82.0,67.0,83.0,73.5);
vector(82.1,67.0,83.0,73.5);
vector(82.2,67.0,83.0,73.5);
vector(82.3,67.0,83.0,73.5);
vector(83.0,84.0,81.0,90.0);
vector(83.0,84.0,81.2,90.0);
vector(83.0,84.0,81.4,90.0);
vector(83.0,84.0,81.6,90.0);
vector(97.0,29.0,95.0,30.0);
vector(95.0,30.0,92.0,33.0);
vector(92.0,33.0,89.0,40.0);
vector(92.0,33.0,89.2,40.0);
vector(92.0,33.0,89.4,40.0);
vector(92.0,33.0,89.6,40.0);
block(82.8,78.0,83.0,80.0);
block(88.5,40.0,90.0,43.0);
block(82.8,57.0,84.0,60.0);
vector(92.0,33.0,89.8,40.0);
block(97.0,29.0,100.0,31.0);
block(86.0,45.0,90.0,51.0);
block(83.0,58.0,86.0,90.0);
block(96.0,29.0,98.0,31.0);
block(95.5,29.5,98.0,31.0);
block(96.5,29.0,98.0,30.0);
vector(97.1,29.0,95.0,30.2);
block(93.5,31.2,96.0,34.0);
block(92.7,32.0,98.0,34.0);
block(90.0,36.6,94.0,41.0);
block(91.0,34.4,94.0,41.0);
block(87.2,42.2,90.0,46.0);
block(84.8,47.2,88.0,51.0);
block(82.0,62.7,85.0,67.0);
block(82.0,88.0,85.0,90.0);
block(82.5,66.0,84.0,71.0);
block(82.5,86.0,84.0,89.0);
block(83.5,53.0,85.0,60.0);
block(82.5,60.0,84.0,65.0);
return;
}

pic6()  {
      /* This subroutine shows how two arrays crcss over
         to cover an area  */
char *st;
```

```
color(1);
block(0.0,90.0,100.0,100.0);
block(0.0,0.0,100.0,28.0);
ocean();
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Hydrophones, Ellipsoids, Quit";
strout(st);
strtxy(4.0,28.5);
st="  A correlation takes place when two different arrays
    cover the same area of";
strout(st);
st="ocean (sector K).  An array picks up the frequency
    changes made from the direction";
strout(st);
st="in which the hydrophone array is pointing and then it
   is compared to the frequencies";
strout(st);
st="obtained by another hydrophone array.  If there is a
    correlation of sounds, then";
strout(st);
st="a possible contact is made somewhere within an ellip-
   soid shaped area near that point.";
strout(st);
color(13);
block(19.6,64.3,21.0,66.0);
color(0);
vector(27.0,33.0,16.0,80.0);
vector(27.0,33.0,19.0,80.0);
vector(80.0,34.1,10.0,72.3);
vector(80.0,34.1,10.0,68.6);
strtxy(22.0,70.0);
st="K";
strout(st);
return;
}

pic62() {
      /* This subroutine shows how the tau line intersection
         will determine the shape of the ellipses   */
char *st;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(1);
block(49.0,29.0,51.0,90.0);
color(7);
vector(5.0,80.0,45.0,35.0);
vector(5.0,35.0,45.0,80.0);
vector(65.0,80.0,80.0,35.0);
vector(80.0,80.0,65.0,35.0);
color(6);
vector(77.0,70.0,77.0,45.0);
```

```
vector(77.0,45.0,76.0,42.0);
vector(76.0,42.0,73.0,41.0);
vector(73.0,41.0,69.8,42.1);
vector(69.8,42.1,68.3,48.0);
vector(68.3,48.0,68.3,70.0);
vector(68.3,70.0,70.2,73.2);
vector(70.2,73.2,73.2,74.4);
vector(73.2,74.4,75.7,73.0);
vector(75.7,73.0,77.0,70.0);
vector(30.9,64.0,33.0,60.0);
vector(33.0,60.0,32.8,54.0);
vector(32.8,54.0,31.0,51.0);
vector(31.0,51.0,27.5,48.5);
vector(27.5,48.5,22.0,49.0);
vector(22.0,49.0,19.3,51.2);
vector(19.3,51.2,17.3,55.3);
vector(17.3,55.3,17.3,60.0);
vector(17.3,60.0,19.3,64.0);
vector(19.3,64.0,23.2,66.2);
vector(23.2,66.2,28.0,66.1);
vector(28.0,66.1,30.9,64.0);
color(2);
strtxy(19.0,85.0);
st="DIAGRAM A                                        DIAGRAM B";
strout(st);
strtxy(10.0,98.0);
st="type Continue, Ellipses combined, Hydrophones, Quit";
strout(st);
strtxy(4.0,28.5);
st="  When a correlation is made from data received at two
    arrays, then an area of probable";
strout(st);
st="contact is determined. The probability that the noise
   came from a specific region of ";
strout(st);
st="water generally forms an ellipse, as in diagrams A and
    B. When the tau lines are almost";
strout(st);
st="perpendicular the the ellipse approximates a circle
    which will have a small area of";
strout(st);
st="probability, however many times the tau lines meet at
    relatively small angles";
strout(st);
st="and cause the area to become a very long ellipse
    (causing high uncertainty).  ";
strout(st);
return;
}

pic621()  {
    /* This subroutine shows how ellipses are combined */
char *st;
pic7();
```

```
pic7a();
color(1);
block(0.0,0.0,100.0,100.0);
strtxy(4.0,97.0);
st="type ==> Continue, Hydrophones, Ellipsoids, Quit";
strout(st);
block(0.0,0.0,100.0,7.0);
return;
}


pic7() {
     /* This subroutine shows how ellipses are combined */
char *st;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(0);
vector(59.7,78.5,62.8,79.6);
vector(62.8,79.6,66.0,80.5);
vector(66.0,80.5,70.2,78.1);
vector(70.2,78.1,72.8,74.0);
vector(72.8,74.0,72.2,68.2);
vector(72.2,68.2,69.0,63.3);
vector(69.0,63.3,46.0,47.5);
vector(46.0,47.5,41.0,45.2);
vector(41.0,45.2,36.4,44.2);
vector(36.4,44.2,33.2,46.2);
vector(33.2,46.2,31.5,50.0);
vector(31.5,50.0,31.8,54.0);
vector(31.8,54.0,33.4,58.2);
vector(33.4,58.2,38.0,63.8);
vector(38.0,63.8,59.7,78.5);
color(15);
vector(46.0,67.8,58.8,64.0);
vector(58.8,64.0,62.0,61.0);
vector(62.0,61.0,62.9,58.0);
vector(62.9,58.0,60.2,55.4);
vector(60.2,55.4,53.3,54.0);
vector(53.3,54.0,42.0,56.8);
vector(42.0,56.8,39.8,60.0);
vector(39.8,60.0,39.6,64.0);
vector(39.6,64.0,42.0,67.0);
vector(42.0,67.0,46.0,67.8);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Quit";
strout(st);
strtxy(4.0,28.5);
st="  When several pieces of data can be gathered about a
    specific frequency, then the";
strout(st);
st="elipses can be combined to reduce the area of probable
```

```
   contact.  As the diagram ";
strout(st);
st="illustrates the second set of data forms an ellipse
    which is inside the first ellipse.";
strout(st);
st="Additional data cause more ellipses to be formed inside
    the present ellipses.";
strout(st);
st="   Type 'm' to see how more ellipses are combined.";
strout(st);
return;
}

pic7a()  {
      /* Draws the combining ellipsoids with a 2 second
         pause between ellipsoid drawings  */
char *st;
color(0);
vector(48.8,64.6,52.2,66.3);
vector(52.2,66.3,55.3,66.0);
vector(55.3,66.0,56.4,64.2);
vector(56.4,64.2,56.0,62.0);
vector(56.0,62.0,54.4,59.8);
vector(54.4,59.8,49.2,56.0);
vector(49.2,56.0,47.0,55.5);
vector(47.0,55.5,45.2,55.4);
vector(45.2,55.4,43.7,57.4);
vector(43.7,57.4,44.3,60.2);
vector(44.3,60.2,48.8,64.6);
sleep(2);
color(15);
vector(52.4,65.7,54.4,65.0);
vector(54.4,65.0,55.2,63.0);
vector(55.2,63.0,54.6,61.2);
vector(54.6,61.2,52.0,58.2);
vector(52.0,58.5,49.0,56.5);
vector(49.0,56.5,47.0,56.0);
vector(47.0,56.0,44.8,56.4);
vector(44.8,56.4,44.0,58.1);
vector(44.0,58.1,45.0,60.2);
vector(45.0,60.2,49.0,64.0);
vector(49.0,64.0,52.4,65.7);
sleep(2);
color(0);
vector(48.3,62.2,50.8,59.8);
vector(50.8,59.8,50.8,58.0);
vector(50.8,58.0,50.0,57.2);
vector(50.0,57.2,48.2,57.0);
vector(48.2,57.0,47.5,57.2);
vector(47.5,57.2,46.0,59.8);
vector(46.0,59.8,46.2,61.5);
vector(46.2,61.5,47.2,62.0);
vector(47.2,62.0,48.3,62.2);
strtxy(48.4,61.0);
```

```
st="A";
strout(st);
strtxy(10.0,40.0);
st="The area around point A is now the contact area.";
strout(st);
return;
}


pic8() {  /* This subroutine shows the overall system */
char *st;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(8);
block(5.0,40.0,6.0,47.0);
block(5.0,50.0,6.0,57.0);
block(5.0,60.0,6.0,67.0);
block(12.0,48.0,25.0,62.0);
block(25.0,75.0,32.0,85.0);
block(40.0,73.0,48.0,80.0);
block(55.0,73.0,75.0,80.0);
block(50.0,45.0,60.0,60.0);
block(72.0,45.0,82.0,60.0);
block(50.0,32.0,75.0,38.0);
vector(5.5,44.0,13.0,55.0);
vector(5.5,54.0,13.0,55.0);
vector(5.5,64.0,13.0,55.0);
vector(20.0,62.0,25.3,80.0);
vector(32.0,80.0,44.0,77.0);
vector(44.0,77.0,55.0,77.0);
vector(65.0,73.0,55.0,60.0);
vector(65.0,73.0,77.0,60.0);
vector(55.0,45.0,65.0,38.0);
vector(77.0,45.0,65.0,38.0);
color(2);
strtxy(3.0,68.0);
st="A";
strout(st);
st="R";
strout(st);
strout(st);
st="A";
strout(st);
st="Y";
strout(st);
st="S";
strout(st);
strtxy(13.0,60.0);
st="  BASE";
strout(st);
st=" STATION";
strout(st);
```

```
strtxy(27.0,82.0);
st="ARC";
strout(st);
strtxy(41.0,78.0);
st="FILTER          CORRELATOR";
strout(st);
strtxy(52.0,58.0);
st=" PEAK           SURFACE";
strout(st);
st=" DATA            PLOTS";
strout(st);
strtxy(55.0,36.0);
st="   TRACKERS";
strout(st);
strtxy(10.0,98.0);
st="type ==> Continue, Techniques (of correlation),
             Parameters, Quit";
strout(st);
strtxy(4.0,28.5);
st=" A brief overview of the entire system demonstrates the
    amount of complex machinery";
strout(st);
st="required to accomplish this surveilance.  Signals are
   recieved at the arrays and are";
strout(st);
st="transmitted to the base station, then sent via satalite
     to the ARC. In the ARC the data";
strout(st);
st="is then sent through filters and a correlator, where
    the computer establishes two sets";
strout(st);
st="of data (peak data files and surface plots). The peak
    data and surface plots are ";
strout(st);
st="reviewed by the trackers to determine any possible
    vessel tracks.";
strout(st);
return;
}

pic5()  {
        /* Same subroutine as in segment 1, but it must be
           redrawn at this point due to the fork process */
char *st;
color(1);
block(0.0,90.0,100.0,100.0);
block(0.0,0.0,100.0,28.0);
ocean();
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Quit";
strout(st);
strtxy(4.0,28.5);
st="  These large distances between arrays allow the system
```

```
     to cover";
strout(st);
st="a large area of ocean, let's say 400 miles by 400 miles
    or roughly";
strout (st);
st="the size of Montana. This area can be further subdivided
    into 100 sectors,";
strout(st);
st="each 40 miles square to make the tracking easier.  Each
    of thes sectors";
strout(st);
st="can presently be covered by XXX correlations per hour.";
strout(st);
disect();
color(13);
block(19.6,64.3,21.0,66.0);
color(0);
st="400 miles";
strtxy(14.0,84.0);
strout(st);
strtxy(27.0,72.0);
strout(st);
return;
}

disect()  {
float x1,x2,y1,y2;
color(0);
x2=26.0;
y1=60.0;
y2=80.0;
for (x1=10.0;x1<=27.0;x1=x1+1.6)
 vector(x1,y1,x1,y2);
x1=10.0;
for (y1=60.0;y1<=81.0;y1=y1+2.0)
 vector(x1,y1,x2,y1);
}     /* End of segment 2  */
```

```
/*##############################################################
# This is the fourth section of the tutorial and it gives #
# the user a detailed look at the types of displays that  #
# he will be seeing under actual operating conditions. It #
# explains specifics to watch out for, whereas the senario#
# gives only a very brief explanation.  The flow of the   #
# program is very similiar to the first two sections.     #
##############################################################*/

main()  {    /* main of the fourth section */
float x,y,z;
char cc,cd,ce,*st;
int samp,l;
ramtek();   /* initializes the Ramtek system */
writon(1);
colortable();   /* establishes a new color table */
colort(10);
pic8();
 cc='z';
while (cc=='z')  {   /* while #1 */
 cc=retchar();
 switch(cc)  {   /* switch #1 */
  case 'c':   /* Continue */
   pic9a();   /* positional data display */
   break;
  case 'r':   /* Return */
   pstop();
    pic8();   /* overall system display */
    cc='z';
   break;
  case 't':  /* technique display */
   pic81();
   cd='z';
   while (cd=='z')  {   /* while #2 */
    cd=retchar();
    switch(cd)  {    /* switch #2 */
     case 'c':   /* Continue */
      pic9a();
      break;
     case 'r':   /* Return */
      pstop();
       pic81();   /* technique display */
       cd='z';
      break;
     case 'f':    /* formula display */
      pic811();
      cc='z';
      break;
     case 'p':    /* parameter display */
      pic82();
      cc='z';
      break;
     default:
```

```
      cd='z';
    }   /* end switch #2 */
   }  /* end while #2 */
   break;
  case 'p':   /* parameter display */
   pic82();
   cc='z';
   break;
  default:
   cc='z';
 }    /* end switch #1 */
}   /* end while #1 */
 cc='z';
 while (cc=='z')  {   /* while #3 */
  cc=retchar();
  switch(cc)  {    /* switch #3 */
   case 'c':   /* Continue */
    pic9b();   /* peak data display */
    break;
   case 'r':   /* Return */
    pstop();
     pic9a();   /* position data display */
     cc='z';
    break;
   default:
    cc='z';
  }   /* end switch #3 */
 }   /* end while #3 */
 cc='z';
 while (cc=='z')  {    /* while #4 */
  cc=retchar();
  switch(cc)  {   /* switch #4 */
   case 'c':   /* Continue */
    pic9c();   /* scores display */
    break;
   case 'r':   /* Return */
    pstop();
     pic9b();   /* peak data display */
     cc='z';
    break;
   default:
    cc='z';
  }   /* end switch #4 */
 }   /* end while #4 */
 cc='z';
 while (cc=='z')  {    /* while #5 */
  cc=retchar();
  switch(cc)  {   /* switch #5 */
   case 'c':   /* Continue */
    pic10();   /* surface plot display */
    break;
   case 'r':   /* Return */
    pstop();
     pic9c();   /* scores display */
```

```
     cc='z';
    break;
   default:
    cc='z';
  }   /* end switch #5 */
 }   /* end while #5 */
 cc='z';
 samp=1;
 while (cc=='z')  {   /* while #6 */
  cc=retchar();
  switch(cc)  {   /* switch #6 */
   case 'c':    /* Continue */
    pic11();    /* covariance matrix display */
    break;
   case 'r':   /* Return */
    pstop();
     pic10();   /* surface plot display */
     cc='z';
    break;
   case 's':   /* sample surface plots */
    if (samp==1)  {
     pic101();
     samp=2;
     }
    else  {
     pic102();
     samp=1;
     }
    cc='z';
    break;
   default:
    cc='z';
  }   /* end switch #6 */
 }   /* end while #6 */
}    /* end main of section 3 */



pstop()  {  /* Quit? subroutine */
char *st,stop,*sp;
sp="  ";
color(11);
block(0.0,0.0,100.0,100.0);
color(5);
strtxy(45.0,80.0);
st="DO  YOU";
strout(st);
strout(sp);
st="REALLY";
strout(st); strout(sp);
st="WANT TO";
strout(st);
strout(sp);
st=" RETURN TO";
```

```
strout(st); strout(sp);
strout("THE MAIN "); strout(sp);
strout("PROGRAM?");
st="(y / n)";
strout(st);
stop=retchar();
if (stop=='y')  {
 color(Ø);
 block(Ø.Ø,Ø.Ø,1ØØ.Ø,1ØØ.Ø);
 exit();
 }
return;
}


colortable()  {   /* establishes a new color table */
 int a[16];
 a[Ø] = triple(Ø,Ø,Ø);       /* black */
 a[1] = triple(15,Ø,Ø);      /* blue */
 a[2] = triple(Ø,15,Ø);      /* green */
 a[3] = triple(Ø,Ø,15);      /* red */
 a[4] = triple(15,5,Ø);      /* light blue */
 a[5] = triple(2,Ø,6);       /* maroon */
 a[6] = triple(Ø,1Ø,Ø);      /* bright green */
 a[7] = triple(2,6,1Ø);      /* peach */
 a[8] = triple(6,1Ø,2);      /* pail green */
 a[9] = triple(1Ø,2,6);      /* purple */
 a[1Ø] = triple (13,5,2);    /* medium blue */
 a[11] = triple(5,2,13);     /* hot pink */
 a[12] = triple (2,13,5);    /* light green */
 a[13] = triple(15,Ø,15);    /* pink */
 a[14] = triple(1Ø,4,4);     /* greyish blue */
 a[15] = triple(15,15,15);   /* white */
 clrtbl(1Ø,a);
 return;
}


pic8() {
     /* Redraws the display of the overall system */
char *st;
color(1);
block(Ø.Ø,Ø.Ø,1ØØ.Ø,1ØØ.Ø);
color(4);
block(Ø.Ø,29.Ø,1ØØ.Ø,9Ø.Ø);
color(8);
block(5.Ø,4Ø.Ø,6.Ø,47.Ø);
block(5.Ø,5Ø.Ø,6.Ø,57.Ø);
block(5.Ø,6Ø.Ø,6.Ø,67.Ø);
block(12.Ø,48.Ø,25.Ø,62.Ø);
block(25.Ø,75.Ø,32.Ø,85.Ø);
block(4Ø.Ø,73.Ø,48.Ø,8Ø.Ø);
block(55.Ø,73.Ø,75.Ø,8Ø.Ø);
block(5Ø.Ø,45.Ø,6Ø.Ø,6Ø.Ø);
```

```
block(72.0,45.0,82.0,60.0);
block(50.0,32.0,75.0,38.0);
vector(5.5,44.0,13.0,55.0);
vector(5.5,54.0,13.0,55.0);
vector(5.5,64.0,13.0,55.0);
vector(20.0,62.0,25.3,80.0);
vector(32.0,80.0,44.0,77.0);
vector(44.0,77.0,55.0,77.0);
vector(65.0,73.0,55.0,60.0);
vector(65.0,73.0,77.0,60.0);
vector(55.0,45.0,65.0,38.0);
vector(77.0,45.0,65.0,38.0);
color(2);
strtxy(3.0,68.0);
st="A";
strout(st);
st="R";
strout(st);
strout(st);
st="A";
strout(st);
st="Y";
strout(st);
st="S";
strout(st);
strtxy(13.0,60.0);
st="  BASE";
strout(st);
st=" STATION";
strout(st);
strtxy(27.0,82.0);
st="ARC";
strout(st);
strtxy(41.0,78.0);
st="FILTER              CORRELATOR";
strout(st);
strtxy(52.0,58.0);
st=" PEAK                SURFACE";
strout(st);
st=" DATA                 PLOTS";
strout(st);
strtxy(55.0,36.0);
st="   TRACKERS";
strout(st);
strtxy(10.0,98.0);
st="type ==> Continue, Techniques (of correlation),
    Parameters, Quit";
strout(st);
strtxy(4.0,28.5);
st=" A brief overview of the entire system demonstrates the
    amount of complex machinery";
strout(st);
st="required to accomplish this surveilance.  Signals are
   recieved at the arrays and are";
```

```
strout(st);
st="transmitted to the base station, then sent via satalite
    to the ARC. In the ARC the data";
strout(st);
st="is then sent through filters and a correlator, where
    the computer establishes two sets";
strout(st);
st="of data (peak data files and surface plots). The peak
    data and surface plots are ";
strout(st);
st="reviewed by the trackers to determine any possible
    vessel tracks.";
strout(st);
return;
}


pic81() {
              /* This display explains the technique used in
                 the correlation process  */
char *st,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,0.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
st="type Continue, Formulas, Parameters, Quit";
strout(st);
sp="   ";
color(5);
st="                     STATEMENT OF THE TECHNIQUE   ";
strtxy(10.0,80.0);
strout(st);
strout(sp);
st="  Given two sequences of time series data, a function
    represents the ";
strout(st);
st="degree of match of the two time series as a function of
    adjustment of the";
strout(st);
st="delay and of adjustment of the relative Doppler ratio
    between the two";
strout(st);
st="series. The term ''coherent'', applied to this function,
    implies that";
strout(st);
st="the relative phase evolutions of the time series are
    made use of in";
strout(st);
st="determining the degree of match, rather than discarded
    as in an incoherent";
strout(st);
st="calculation.  The time series of interest to us are
```

```
    finite, discrete,";
strout(st);
st="frequency-shifted, band-limited, complex time series
    obtained from";
strout(st);
st="passive sonar sensors (hydrophone arrays).";
strout(st);
return;
}


pic811() {
     /* This display illustrates the formulas used
        in the correlation  */
char *st,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,25.0,100.0,90.5);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Techniques (of correlation),
     Parameters, Quit";
strout(st);
sp="   ";
color(5);
strtxy(5.0,88.0);
st="1)  X(a,t) = ---      X'(n)  X (n-t/  T)e";
strout(st);
strout(sp);
st="2)  X'(n) = X (an-(a-1)N/2)";
strout(st);
strout(sp);
st="3)  A =      X'(n) X (n)      X (m-t/  T)X (m-t/  T)";
strout(st);
strout(sp);
st=" where  X(a,t)  is the modulus correlation coefficient";
strout(st);
st="          a is the relative Doppler ratio compensation";
strout(st);
st="          t is the delay compensation";
strout(st);
st="          A is the normalization constant";
strout(st);
st="X  and X  are time series sampled at time intervals T";
strout(st);
st="          F1 and F2 are the center frequency of the
      passband for the time series";
strout(st);
strtxy(5.0,36.4);
st="           1      2";
strout(st);
strtxy(5.0,90.2);
st="               1  N-1            *     2lli(aF1-F2)n  T";
```

```
strout(st);
st="                A  n=o   1       2 ";
strout(st);
strout(sp);
st="    1        1";
strout(st);
st="    2  N-1          *    N-1                   * ";
strout(st);
st="           n=o   1    1    m=o   2             2";
strout(st);
vector(9.0,84.5,9.0,88.2);
vector(15.8,84.5,15.8,88.2);
vector(23.0,82.0,23.0,90.0);
vector(68.0,82.0,68.0,90.0);
block(35.0,86.0,35.5,86.8);
vector(45.0,85.0,46.5,85.0);
vector(45.0,85.0,45.7,87.0);
vector(45.7,87.0,46.5,85.0);
vector(64.5,87.0,66.0,87.0);
vector(64.5,87.0,65.2,89.0);
vector(65.2,89.0,66.0,87.0);
vector(51.0,90.2,53.0,90.2);
vector(45.8,65.0,47.3,65.0);
vector(45.8,65.0,46.5,67.0);
vector(46.5,67.0,47.3,65.0);
vector(58.2,67.0,59.0,65.0);
vector(57.5,65.0,58.2,67.0);
vector(57.5,65.0,59.0,65.0);
vector(13.5,67.5,17.0,67.5);
vector(13.5,67.5,16.0,66.2);
vector(13.5,65.0,16.0,66.2);
vector(13.5,65.0,17.0,65.0);
vector(32.0,67.5,35.5,67.5);
vector(32.0,67.5,34.0,66.2);
vector(32.0,65.0,34.0,66.2);
vector(32.0,65.0,35.5,65.0);
block(24.8,66.4,25.2,66.9);
vector(24.0,87.0,27.5,87.0);
vector(24.0,87.0,26.0,85.8);
vector(24.0,84.5,26.0,85.8);
vector(24.0,84.5,27.5,84.5);
vector(71.7,35.0,73.3,35.0);
vector(71.7,35.0,72.5,37.0);
vector(72.5,37.0,73.3,35.0);
strtxy(4.0,22.5);
color(2);
st="The basic correlation structure can be seen in equation
     1 as the modulus of";
strout(st);
st="normalized sum of products (with one series conjugated)
    of the time series";
strout(st);
st="samples. The other two equations are used for delay and
    Doppler compensation";
```

```
strout(st);
st="of the time series prior to correlation.";
strout(st);
return;
}


pic82() {
      /* This display shows the various parameters that
         can be altered to get a better picture  */
char *st,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Techniques (of correlation), Quit";
strout(st);
strtxy(16.0,25.0);
st="  Listed above are a few of the various parameters that
    can be";
strout(st);
st="adjusted to get a better track of the vessel.  Other
    parameters";
strout(st);
st="are demonstrated throughout the tutorial and if the
    limits are";
strout(st);
st="exceeded, the computer will adjust and notify you.";
strout(st);
color(0);
vector(43.0,76.0,51.5,76.0);
strtxy(20.0,80.0);
st="                      RANGES";
strout(st);
st="LONGITUDE           30.00-50.00";
strout(st);
st="LATIDUDE           150.00-130.00";
strout(st);
st="COHERENCE LIMIT      0.0-2.0";
strout(st);
st="TRACK NUMBER           1-10";
strout(st);
st="RUN NUMBER             1-7";
strout(st);
st="REGION NUMBER          1-25";
strout(st);
return;
}

pic9a() {
          /* This display shows the user what kind of
             information he will receive when he actually
```

```
                uses the system    */
char *st,cc,*st1,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Quit";
strout(st);
color(5);
strtxy(5.0,86.0);
st1="     RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9";
strout(st1);
sp="   ";
strout(sp);
   /* Shows the position, course, and speed of a vessel  */
st="Water Time    Peak    Position Coordinates    Courses
strout(st);
st=" 23-Feb-80    Number   Latitude  Longitude     Course
strout(st);
strout(sp);
st=" 14:58:03        0     39.5000   147.5000        0.0
strout(st);
st=" 14:58:03     3427     39.4711   147.4707      142.0
strout(st);
st=" 15:04:23     3533     39.4682   147.4552      154.7
strout(st);
st=" 15:14:30     3449     39.4212   147.4189      153.9
strout(st);
st=" 15:38:35     3540     39.3582   147.4417      160.3
strout(st);
st=" 15:48:44     3452     39.3980   147.4815      161.6
strout(st);
color(2);
strtxy(4.0,28.5);
st="  This data illustrates the estimated position, course
    and speed of a suspected";
strout(st);
st="vessel.  Depending on how consistent the data is, the
    more accurate the estimated";
strout(st);
st="position probably is.  For example, the courses and
    speeds in the above display";
strout(st);
st="are relatively close together, thus the data is
    probably from a vessel.  If the";
strout(st);
st="courses were 159.0, 135.6, 160.4, 122.7, 150.3 then the
   data would probably be";
strout(st);
st="from noise in the ocean.";
strout(st);
}
```

```
pic9b() {
   /* Shows the reference stations and their tau and
      doppler values  */
char *st,cc,*st1,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Quit";
strout(st);
color(5);
strtxy(5.0,86.0);
st1="    RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9";
strout(st1);
sp="  ";
strout(sp);
st="Receipt Time   Peak   -REF Station-   -PRD Station-
    ------Peak Data-------";
strout(st);
st=" 20-Feb-80    Number   STM/BM   Freq    STM/BM   Freq
      TAU    DOPPLER  Gamma-Sq";
strout(st);
strout(sp);
st=" 00:00:00         0           0   0.00          0   0.00
strout(st);
st=" 15:16:15      3427    1323024  41.40   1322029  41.40
strout(st);
st=" 15:16:15      3533    1322031  41.40   2411020  41.00
strout(st);
st=" 15:33:19      3449    1323024  41.40   1322030  41.40
strout(st);
st=" 15:50:23      3540    1322031  41.40   2411021  41.40
strout(st);
st=" 16:07:27      3453    1323025  41.40   1322030  41.40
strout(st);
color(2);
strtxy(4.0,28.5);
st="  This data gives you the same peak numbers, the array
    stations that received the";
strout(st);
st="signals, the frequency of the signal and peak data. The
   variances in the tau and";
strout(st);
st="doppler values can give evidence as to the validity of
   the track. The switching of";
strout(st);
st="the tau values from -416.7 to 27.3 to -412.8 ect. means
    that the ellipsoids will ";
strout(st);
st="probably combine very well because the data obviously
    is being collected at two ";
```

```
strout(st);
st="different sets of arrays.";
strout(st);
return;
}

pic9c() {
    /* Shows the Chi-square scores for the various data */
char *st,cc,*st1,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Quit";
strout(st);
color(5);
strtxy(5.0,86.0);
st1="    RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9";
strout(st1);
sp="  ";
strout(sp);
st="Water Time      peak       Prob       --CHI Square Scores--
strout(st);
st=" 23-Feb-80    Number      Score    Stagewise   Cumulative
strout(st);
strout(sp);
st=" 14:58:03          0       0.00      0.0000       0.0000
strout(st);
st=" 14:58:03       3427      -1.80      0.3171       0.3171
strout(st);
st=" 15:04:23       3533      -4.89      0.0162       0.1667
strout(st);
st=" 15:14:30       3449      -3.99      0.4410       0.2581
strout(st);
st=" 15:38:35       3540      -4.66      1.4096       0.2460
strout(st);
st=" 15:48:44       3453      -4.81      2.0796       0.8527
strout(st);
color(2);
strtxy(4.0,28.5);
st="  This data shows the same peaks, their Chi Square
    probabilities of actual ";
strout(st);
st="position, and their measured residuals.  The Chi Square
    scores are broken ";
strout(st);
st="into two groups, each individual score and the cumula-
    tive total.  A good ";
strout(st);
st="check as to the probability of an actual contact is the
    cumulative score.";
strout(st);
```

```
st="near 1.0 is quite good.";
strout(st);
return;
}


pic10() {
                /* This display explains what a surface
                   plot will look like and show.  */
char *st,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Samples, Quit";
strout(st);
strtxy(4.0,28.5);
st="  By examining a surface plot the tracker may be able
    to see a pattern of peaks";
strout(st);
st="that would establish a vessel's track.  Care must be
   taken when viewing the";
strout(st);
st="surface plot displays because the size of the area
   (range of latitude and longitude)";
strout(st);
st="can be varied, thus giving the appearance of a track
    when there might not be";
strout(st);
st="one there.";
strout(st);
color(7);
block(40.0,45.0,71.5,78.0);
color(0);
st="Region Number 6";
strtxy(5.0,70.0);
strout(st);
st="Coherence Limits 1.0";
strout(st);
st="--";
sp="  ";
strtxy(38.5,80.7);
strout(sp);
strout(st);
strout(sp);
strout(st);
strout(sp);
strout(st);
strout(sp);
st=" 1      1      1      1      1";
strout(st);
st="40.00";
```

```
strtxy(32.5,75.7);
strout(st):
strtxy(32.5,46.2);
st="38.ØØ --";
strout(st);
strtxy(36.Ø,42.7);
st=" 147.ØØ                       145.ØØ";
strout(st);
circ();
return;
}

circ()  {  /* Draws a surface plot */
char *A,*S,*D,*F,*G,*H,*J,*K,*L;
A="O     o      O   o  o     o  O";
S="   o      o     O     o    o  ";
D="o          O      o          o";
F="     O         O         o    ";
G="         oo         o    O    ";
H="  o       O           o     o ";
J="             o   O      O     ";
K="   o     o         O  o    o  ";
L="  o   O         o        O    ";
color(Ø);
strtxy(4Ø.Ø,8Ø.Ø);
strout(A);
strout(S);
strout(D);
strout(F);
strout(G);
strout(H);
strout(J);
strtxy(4Ø.Ø,77.Ø);
dblwid(1);
strout(L);
strout(A);
strout(H);
strout(H);
strout(H);
strtxy(41.Ø,59.9);
dblwid(Ø);
strout(D);
strout(K);
strout(L);
dblwid(Ø);
color(4);
block(39.Ø,78.Ø,1ØØ.Ø,85.Ø);
block(71.5,44.Ø,1ØØ.Ø,85.Ø);
}


pic1Ø1() {  /*Draws another surface plot sample */
char *st,*sp;
color(1);
```

```
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Samples, Quit";
strout(st);
sp=" ";
strtxy(4.0,28.5);
strout(sp);
st="This surface plot shows a very probable track, pointed
    to by the arrow.";
strout(st);
st="Because the size of the plot is relatively small (60
     miles square) and the";
strout(st);
st="coherence limit is relatively high ( 1.0 ), this is
    probably a track of a";
strout(st);
st="vessel and should be further investigated.";
strout(st);
color(7);
block(40.0,45.0,71.5,78.0);
color(0);
st="Region Number 23";
strtxy(5.0,70.0);
strout(st);
st="Coherence Limits 1.0";
strout(st);
st="--";
sp="  ";
strtxy(38.5,80.7);
strout(sp);
strout(st);
strout(sp);
strout(st);
strout(sp);
strout(st);
strout(sp);
st=" 1      1      1      1      1";
strout(st);
st="39.00";
strtxy(32.5,75.7);
strout(st);
strtxy(32.5,46.2);
st="38.00 --";
strout(st);
strtxy(36.0,42.7);
st=" 147.00                      146.00";
strout(st);
circ2();
color(0);
block(50.4,80.0,51.0,87.0);
vector(50.7,79.0,49.4,83.0);
```

```
vector(50.7,79.0,52.0,83.0);
return;
}

circ2()  {    /* Drawing a surface plot  */
char n;
char *sp,*st;
char *A,*S,*I,*F,*G,*H,*J,*K,*L;
sp=" ";
A="O    o      O   c  o     o  O";
S="  o      o           o       ";
D="o         O      o          o";
F="    O         O         o    ";
G="        oc         o    O    ";
H=" o       O           o     o ";
J="            o   O      O     ";
K="  o     o         O  c    o  ";
L="     O         o        O    ";
color(0);
strtxy(40.0,80.0);
strout(A);
strout(S);
strout(D);
strout(F);
strout(G);
strout(H);
strout(J);
strtxy(40.0,77.0);
dblwid(1);
strout(L);
strout(A);
strout(H);
strout(L);
strout(L);
strtxy(41.0,59.9);
dblwid(0);
strout(D);
strout(K);
strout(L);
dblwid(0);
strtxy(42.0,72.0);
strout(G);
strout(G);
strout(S);
strout(G);
strtxy(40.0,76.0);
strout(G);
strout(S);
strout(A);
strtxy(40.0,75.0);
dblwid(1);
strout(K);
strout(F);
strout(H);
```

```
strout(L);
st=" Oo";
strtxy(47.0,82.0);
strout(st);
strout(st);
strout(sp);
strout(st);
strout(sp);
strout(st);
strout(st);
color(4);
dblwid(0);
block(39.0,78.0,100.0,85.0);
block(71.5,44.0,100.0,85.0);
}


pic11() {
       /* This display shows the covariance matrix and then
          asks the user if he wishes to move on to the
          Senario or redo the tutorial  */
char *st,cc,*st1,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
st="type Continue, Senario, Tutorial (again), Quit";
strout(st);
strtxy(5.0,88.0);
color(5);
st1="    RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9";
strout(st1);
sp="  ";
st="Water Time     peak     ----------------COVARIANCE
     MATRIX--------------   ";
strout(st);
st=" 23-Feb-80    Number  ";
strout(st);
strtxy(5.5,73.0);
st=" 14:58:03      3427         0.03136       0.00000      ";
strout(st);
st="                            0.00000     37.88962
strout(st);
st="                           -0.03150      0.00000
strout(st);
st="                            0.00000    -48.49765
strout(st);
strtxy(5.5,49.0);
st=" 15:04:23      3533         0.00985      0.00047      ";
strout(st);
st="                            0.00047      0.31107
strout(st);
```

```
st="                                    0.00551       0.00178
strout(st);
st="                                   -0.00001       0.19371
strout(st);
color(2);
strtxy(4.0,28.5);
strout(sp);
st="  This display illustrates the covariance matrices which
     can be used to";
strout(st);
st="determine how accurate the estimated courses and speeds
    are.  An experienced";
strout(st);
st="tracker could determine this by examining each 4 X 4
     matrix, however this is";
strout(st);
st="difficult and thus will be handled by the computer.";
strout(st);
return;
}


pic102() {
     /* This subroutine draws another surface plot */
char *st,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,29.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
st="type ==> Continue, Samples, Quit";
strout(st);
color(7);
block(40.0,45.0,71.5,78.0);
color(0);
st="Region Number 3";
strtxy(5.0,70.0);
strout(st);
st="Coherence Limits 0.7";
strout(st);
st="--";
sp="  ";
strtxy(38.5,80.7);
strout(sp);
strout(st);
strout(sp);
strout(st);
strout(sp);
strout(st);
strout(sp);
st=" 1      1      1      1      1";
strout(st);
st="40.00";
```

```
strtxy(32.5,75.7);
strout(st);
strtxy(32.5,46.2);
st="33.00 --";
strout(st);
strtxy(36.0,42.7);
st=" 147.00                              140.00";
strout(st);
color(0);
circ();
circ2();
circ3();
return;
}

circ3()  {
char *Q,*W,*E,*R,*T;
Q="    oo       O      ";
W=" o  O     o    o    ";
E="  O       Oo    O   ";
R="    o       o  O    ";
T=" O  O       O       ";
strtxy(40.3,79.8);
color(0);
strout(Q);
strout(W);
strout(E);
strout(R);
strout(T);
strtxy(40.9,81.0);
strout(E);
strout(R);
strout(E);
strout(E);
strout(R);
dblwid(1);
strout(W);
strtxy(40.0,78.0);
strout(T);
strout(Q);
strout(W);
strout(W);
strout(R);
strtxy(41.0,78.7);
strout(E);
strout(W);
strout(E);
strout(T);
strout(R);
dblwid(0);
color(4);
block(39.0,78.0,100.0,85.0);
block(71.5,44.0,100.0,85.0);
return;    }
```

```
/*################################################################
#  This section of code is designed to give the user prac- #
# tice visualizing the types of data that will be presented#
# in the actual tracking. It is basically set up to run    #
# four types of data depending on the parameters that the  #
# user tries to use. Each display gives the examples and   #
# then tells the user what the problem areas might be, if  #
# there are any. This is the fifth section of the main     #
# program but can be called up by itself by typing Senerio #
################################################################*/



int choice,cont,run,region,track,coher,lath,latl,longh,longl;
char *srun,*sregion,*strack,*scoher,*slath,*slatl,*slongh;
char *N,*B,*M,*Q,*W,*E,*R,*T,*Y,*slongl,*top;
       /*  Setting global values because they are used in
           many of the displays   */


main()  {   /* Senerio section of program */
char cc;
ramtek();   /* initializing the Ramtek subroutines  */
writon(1);
changcolor();  /* Establishing a new color table  */
colort(10);
top="type==>Menu, Location, Peak, Accuracy, Surface, Quit";
    /* Setting a global string that gets displayed at the
       top of the majority of the displays   */
cont=1;
sregion="10";
srun="6";
strack="2";
scoher="1.0";
slath="35.45";  /* initializing global values */
slatl="35.00";
slongh="125.45";
slongl="125.00";
senm();
while (cont==1)  {   /* while #1 */
 cc=retchar();
 switch(cc)  {   /* switch #1 */
     /* Switching of the displays dependent on the desires
        of the user at the keyboard   */
  case 'm':   /* displays the menu  */
   senm();
   break;
  case 'p':   /* displays peak data */
   senp();
   break;
  case 'l':   /* displays positional data */
   senl();
   break;
```

```
  case 'a':  /* displays the accuracy data */
   sena();
   break;
  case 's':   /* displays the surface plots */
   sens();
   break;
  case 'r':   /* Return */
   pstop();
   break;
  default:
   cont=1;
  }   /* end of switch #1 */
 }   /* end of while #1 */
}    /* End of Main  */
```

```
senm()  {     /* This display presents a menu of parameters
                 that can be altered by the user to get a
                 better view of the situation  */
char *st,*sp;
color(7);
block(0.0,0.0,100.0,100.0);
color(10);
block(0.0,90.0,100.0,100.0);
color(5);
strtxy(10.0,97.0);
st="type in the desired numbers or zeros.";
strout(st);
strtxy(6.0,86.0);
st="  There are a number of parameters that you can change
     to survey different tracks,";
strout(st);
st="runs, and regions.  The limits are listed along with the
    present value of the";
strout(st);
st="parameter.  When the block appears to the right of the
    present value, just type";
strout(st);
st="in the value desired followed by <cr> (if the present
    value is okay type  0 <cr>).";
strout(st);
sp="   ";
strout(sp);
strtxy(15.0,58.0);
st="    LIMIT          PARAMETER       PRESENT VALUE
     NEW VALUE";
strout(st);
strout(sp);
st="     1-25           REGION                 ";
strout(st);
st="     1-7              RUN                  ";
strout(st);
st="     1-10            TRACK                 ";
```

```
strout(st);
st="   0.1-2.0          COHERENCE              ";
strout(st);
st=" 30.00-50.00     LATITUDE(high)            ";
strout(st);
st=" 30.00-50.00     LATITUDE(low)             ";
strout(st);
st="120.00-140.00    LONGITUDE(high)           ";
strout(st);
st="120.00-140.00    LONGITUDE(low)            ";
strout(st);
strtxy(57.0,48.5);
strout(sregion);/* Prints out present values of parameters*/
strout(srun);
strout(strack);
strout(scoher);
strout(slath);
strout(slatl);
strout(slongh);
strout(slongl);
newval();        /* calls the subroutine that checks the
                    values being inserted by the user  */
senma();   /* calls for a new display  */
if (coher<=07)  {   /* Establishes what set of displays will
                       be presented based on the values that
                       were just inputted by the user  */
 if ((lath-latl) <= 100)
  choice='c';
 else
  choice='b';
 }
else  {
 if ((lath-latl) <= 100)
  choice='d';
 else
  choice='a';
}
}


senma()  {   /* This displays the basic parameters and then
                allows the user to continue, quit, or change
                the parameters.   */
char *sp,cc,*st;
color(7);
block(0.0,0.0,100.0,100.0);
color(10);
block(0.0,90.0,100.0,100.0);
color(5);
strtxy(5.0,97.0);
strout("type ==> Continue, Menu, Quit");
sp="   ";
strtxy(25.0,60.0);
st="RUN NUMBER   ";
```

```
strout(st);
strout(sp);
st="REGION NUMBER    ";
strout(st);
strout(sp);
st="TRACK NUMBER    ";
strout(st);
strout(sp);
st="COHERENCE LIMIT    ";
strout(st);
cc='z';
while (cc=='z') {
 cc=retchar();
 switch(cc)  {
  case 'c':   /* Continue */
   senl();
   break;
  case 'm':   /* displays the menu */
   senm();
   break;
  case 'q':   /* Quit */
   pstop();
   break;
  default:
   cc='z';
  }
}
}


pstop() {    /* This display is presented anytime the user
                types 'q' and then finds out if he really
                wants to quit.    */
char *st,*sp,cb;
color(3);
block(0.0,0.0,100.0,100.0);
color(2);
sp="  ";
strtxy(46.0,80.0);
st="DO  YOU";
strout(st);
strout(sp);
st="REALLY";
strout(st);
strout(sp);
st="WANT TO";
strout(st);
strout(sp);
st="RETURN TO";
strout(st);
strout(sp);
strout("THE MAIN"); strout(sp);
strout("PROGRAM?");
st="(y / n)";
```

```
strout(st);
cb=retchar();
if(cb=='y')  {
 cont=0;  /* Sets global value to quit senario */
 color(0);
 block(0.0,0.0,100.0,100.0);
}
else
 senl();
}


newval()  {   /* This subroutine receives the inputs of the
                 user for the changing of the parameters,
                 evaluates them, rejects any that exceed
                 the limits, and then inserts decimal points
                 where necessary.    */
int t,tcoher,tregion,trun,ttrack,tlath,tlatl,tlongh,tlongl;
t=0;
while (t==0)  {
color(15);
 block(73.0,45.0,88.0,50.0);
 strtxy(74.0,49.0);
color(5);
 tregion=getnum(10);  /* changes the region number */
 if (tregion != 0)  {
  if ((tregion <= 25)&&(tregion >= 1))  {
   itoa(sregion,tregion);
   region=tregion;
   t=1;
  }
 else
  t=0;
 }
 else
  t=1;
}
t=0;
while (t==0)  {
color(15);
 block(73.0,39.5,88.0,45.0);
 strtxy(74.0,43.0);
color(5);
 trun=getnum(10);  /* changes the run number  */
 if (trun!=0)  {
  if ((trun<=7)&&(trun>=1))  {
   itoa(srun,trun);
   run=trun;
   t=1;
   }
  else
   t=0;
  }
 else
```

```
  t=1;
}
t=0;
while(t==0)  {
color(15);
 block(73.0,34.0,88.0,39.0);
 strtxy(74.0,38.0);
color(5);
 ttrack=getnum(10);  /* changes the track number */
 if(ttrack!=0)  {
  if((ttrack<=10)&&(ttrack>=1))  {
   itoa(strack,ttrack);
   track=ttrack;
   t=1;
   }.
  else
   t=0;
  }
 else
  t=1;
}
t=0;
while(t==0)  {
color(15);
 block(73.0,27.5,88.0,33.5);
 strtxy(74.0,33.2);
color(5);
 tcoher=getnum(10);  /* changes the coherence limit */
 if (tcoher != 0)  {
  if((tcoher<=20)&&(tcoher>=01))  {
   itoa(scoher,tcoher);
   coher=tcoher;
   scoher[4]=scoher[3];
   scoher[3]=scoher[2];
   scoher[2]='.';  /* inserts a decimal point */
   t=1;
   }
  else
   t=0;
  }
 else
  t=1;
}
t=0;
while(t==0)  {
color(15);
 block(73.0,23.0,88.0,28.0);
 strtxy(74.0,28.0);
color(5);
 tlath=getnum(10);  /* changes the high latitude value */
 if(tlath!=0)  {
  if((tlath<=5000)&&(tlath>3000))  {
   itoa(slath,tlath);
   lath=tlath;
```

```
    slath[6]=slath[5];
    slath[5]=slath[4];
    slath[4]=slath[3];
    slath[3]='.';  /* inserts a decimal point */
    t=1;
    }
   else
    t=0;
   }
  else
   t=1;
 }
 t=0;
 while(t==0)  {
 color(15);
  block(73.0,17.5,88.0,23.5);
  strtxy(74.0,23.3);
 color(5);
  tlatl=getnum(10);  /* changes the low latitude value */
  if(tlatl!=0)  {
   if((tlatl<5000)&&(tlatl>=3000)&&(tlatl<tlath))  {
     /* Makes sure that latitude low is less than the
        high latitude   */
    itoa(slatl,tlatl);
    latl=tlatl;
    slatl[6]=slatl[5];
    slatl[5]=slatl[4];
    slatl[4]=slatl[3];
    slatl[3]='.';  /* inserts a decimal point  */
    t=1;
    }
   else
    t=0;
   }
  else
   t=1;
 }
 t=0;
 while(t==0)  {
 color(15);
  block(73.0,12.0,88.0,18.2);
  strtxy(74.0,18.0);
 color(5);
  tlongh=getnum(10);  /*Changes the high longitude value*/
  if(tlongh!=0)  {
   if((tlongh<=14000)&&(tlongh>12000))  {
    itoa(slongh,tlongh);
    longh=tlongh;
    slongh[6]=slongh[5];
    slongh[5]=slongh[4];
    slongh[4]=slongh[3];
    slongh[3]='.';  /* inserts a decimal point  */
    t=1;
    }
```

```
  else
   t=0;
  }
 else
  t=1;
}
t=0;
while(t==0)  {
color(15);
 block(73.0,8.5,88.0,11.5);
 strtxy(74.0,13.0);
color(5);
 tlongl=getnum(10);  /* changes the low longitude value */
 if(tlongl!=0)  {
  if((tlongl<=14000)&&(tlongl>=12000)&&(tlongl<tlongh))  {
          /* Makes sure that the value of low longitude is
             less than that of high longitude   */
   itoa(slongl,tlongl);
   longl=tlongl;
   slongl[6]=slongl[5];
   slongl[5]=slongl[4];
   slongl[4]=slongl[3];
   slongl[3]='.';  /* inserts a decimal point  */
   t=1;
   }
  else
   t=0;
  }
 else
  t=1;
}
}

itoa(ptr,no)   /* This subroutine changes an interger
                  into a string of characters  */
 char *ptr;
 int no;
{
int K;
if((K=no/10) != 0)
 ptr = itoa(ptr,K);
*ptr=no + '0';
*++ptr = '\0';
return(ptr);
}



senp() {  /* This displays the peak data */
char *st,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,15.0,100.0,90.0);
```

```
color(2);
strtxy(10.0,98.0);
strout(top);
color(5);
strtxy(5.0,86.0);
st="      RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9";
strout(st);
sp="  ";
strout(sp);
st="Receipt Time   Peak    -REF Station-   -PRD Station-
    ------Peak Data-----";
strout(st);
st=" 20-Feb-80    Number   STM/BM   Freq    STM/BM   Freq
    TAU    DOPPLER  Gamma-Sq";
strout(st);
strout(sp);
switch(choice)  {       /* There are four possible sets of data
                           can be displayed dependent on the
                           various limits set by the user */
 case 'd':
st=" 00:00:00          0           0   0.00          0   0.00
strout(st);
st=" 15:16:15      3427    1323024  48.70   1322029  48.70
strout(st);
st=" 15:16:15      3533    1322031  48.70   2411020  48.70
strout(st);
st=" 15:33:19      3449    1323024  48.70   1322030  48.70
strout(st);
st=" 15:50:23      3540    1322031  48.70   2411021  48.70
strout(st);
st=" 16:07:27      3453    1323025  48.70   1322030  48.70
strout(st);
st=" 16:08:31      3421    1322032  48.70   2411020  48.70
strout(st);
st=" 16:09:56      3479    1323024  48.70   1322031  48.70
strout(st);
st=" 16:11:34      3429    1322031  48.70   2411023  48.70
strout(st);
strtxy(4.0,12.0);
st="GOOD TRACK WITH TWO SETS OF ARRAYS SWITCHING BACK AND
    FORTH TO PROVIDE DATA.";
strout(st);
st="( NOTICE THE TAU VALUES ALTERNATING )";
strout(st);
break;
case 'a':
st=" 16:52:56          0           0  48.70          0  48.70
strout(st);
st=" 16:52:56      3542    1333031  48.70   1346024  48.70
strout(st);
st=" 16:55:37      3478    1333041  48.70   1346031  48.70
strout(st);
st=" 16:58:10      3548    1333027  48.70   1346028  48.70
strout(st);
```

```
st=" 16:59:19       3557     1322107  48.70    2411321  48.70
strout(st);
st=" 17:03:02       3461     1333034  48.70    1346021  48.70
strout(st);
st=" 17:05:56       3640     1333051  48.70    1346039  48.70
strout(st);
st=" 17:09:42       3582     1322114  48.70    2411324  48.70
strout(st);
st=" 17:11:48       3533     1333029  48.70    1346042  48.70
strout(st);
strtxy(4.0,12.0);
st="NOT MUCH ALTERNATING OF ARRAY PAIRS, THUS PROBABLY NOT
    TOO GOOD A TRACK.";
strout(st);
break;
case 'b':
st=" 21:36:04          0           0  48.70          0  48.70
strout(st);
st=" 21:36:04       3560     2411031  48.70    1441024  48.70
strout(st);
st=" 21:37:09       3532     2412034  48.70    1442067  48.70
strout(st);
st=" 21:37:52       3571     2411041  48.70    1441028  48.70
strout(st);
st=" 21:38:38       3564     2411038  48.70    1441021  48.70
strout(st);
st=" 21:39:41       3551     2411036  48.70    1441025  48.70
strout(st);
st=" 21:40:02       3554     2411028  48.70    1441020  48.70
strout(st);
st=" 21:40:58       3584     2411031  48.70    1441024  48.70
strout(st);
st=" 21:41:44       3547     2411035  48.70    1441018  48.70
strout(st);
strtxy(4.0,12.0);
st="NO SWITCHING OF ARRAY PAIRS, YET THERE IS A LOT OF
    VARIANCE IN THE TAU VALUES,";
strout(st);
st="THUS A VERY POOR TRACK.";
strout(st);
break;
case 'c':
st=" 06:12:37          0           0  48.70          0  48.70
strout(st);
st=" 06:12:37       3325     1341021  48.70    1472365  48.70
strout(st);
st=" 06:13:25       3342     1834261  48.70    2437001  48.70
strout(st);
st=" 06:14:03       3394     1341036  48.70    1472341  48.70
strout(st);
st=" 06:14:57       3327     1834275  48.70    2437009  48.70
strout(st);
st=" 06:16:32       3319     1341028  48.70    1472355  48.70
strout(st);
```

```
st=" 06:17:29      3364    1341025  48.70   1472351  48.70
strout(st);
st=" 06:18:54      3318    1834268  48.70   2437005  48.70
strout(st);
st=" 06:20:21      3343    1341029  48.70   1472368  48.70
strout(st);
strtxy(4.0,12.0);
st="FAIRLY GOOD SWITCHING OF ARRAY PAIRS BUT WITHIN AN ARRAY
    PAIR THERE IS TOO MUCH";
strout(st);
st="VARIANCE DUE TO THE LOW COHERENCE LIMITS.";
strout(st);
break;
}
}

senl() {        /* This subroutine displays the position,
                   course, and speed of the contact  */
char *st,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,15.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
strout(top);
color(5);
strtxy(5.0,86.0);
st="      RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9";
strout(st);
sp="  ";
strout(sp);
st="Water Time    Peak    Position Coordinates   Courses and
    Speeds";
strout(st);
st=" 23-Feb-80    Number   Latitude  Longitude     Course
    Speed";
strout(st);
strout(sp);
switch(choice)  {
      /* Displays one of four possible sets of data */
case 'a':
st=" 16:52:56        0      38.5000   142.5000        0.0
strout(st);
st=" 16:52:56     3542      38.4731   142.4873      142.0
strout(st);
st=" 16:55:37     3478      38.4422   142.4791      150.6
strout(st);
st=" 16:58:10     3548      38.4431   142.4762      147.3
strout(st);
st=" 16:59:19     3557      38.4401   142.4750      156.4
strout(st);
st=" 17:03:02     3461      38.4324   142.4775      156.8
strout(st);
```

```
st=" 17:05:56      3640      38.4246   142.4689       144.7
strout(st);
st=" 17:09:42      3582      38.4263   142.4663       136.2
strout(st);
st=" 17:11:48      3533      38.4221   142.4572       149.7
strout(st);
strtxy(4.0,12.0);
st="POSSIBLY A GOOD TRACK, BUT THE SIZE OF THE REGION BEING
    COVERED IS A LITTLE";
strout(st);
st="TOO LARGE TO GET A REAL GOOD PICTURE OF THE SITUATION.";
strout(st);
break;
case 'b':
st=" 21:36:04         0      36.5000   143.5000         0.0
strout(st);
st=" 21:36:04      3560      36.4784   143.6534       020.5
strout(st);
st=" 21:37:09      3532      36.5348   143.5739       034.6
strout(st);
st=" 21:37:52      3571      36.5782   143.4936       025.9
strout(st);
st=" 21:38:38      3564      36.6139   143.4467       012.3
strout(st);
st=" 21:39:41      3551      36.6317   143.4052       005.3
strout(st);
st=" 21:40:02      3554      36.6308   143.3761       014.7
strout(st);
st=" 21:40:58      3584      36.6536   143.3485       018.1
strout(st);
st=" 21:41:44      3547      36.6738   143.3263       023.5
strout(st);
strtxy(4.0,12.0);
st="POOR TRACK AS SEEN BY THE COURSE AND SPEED CHANGES,
    PROBABLY DUE TO THE";
strout(st);
st="LARGE AREA AND THE LOW COHERENC LEVEL.";
strout(st);
break;
case 'c':
st=" 06:12:37         0      33.5000   145.5000         0.0
strout(st);
st=" 06:12:37      3325      33.4734   145.5398       228.4
strout(st);
st=" 06:13:25      3342      33.4689   145.5402       235.6
strout(st);
st=" 06:14:03      3394      33.4502   145.5693       243.5
strout(st);
st=" 06:14:57      3327      33.4294   145.5728       227.9
strout(st);
st=" 06:16:32      3319      33.3825   145.5829       222.5
strout(st);
st=" 06:17:29      3364      33.3518   145.5962       225.3
strout(st);
```

```
st=" 06:18:54     3318     33.3329   145.6036      233.1
strout(st);
st=" 06:20:21     3343     33.2185   145.6285      242.3
strout(st);
strtxy(4.0,12.0);
st="EVEN THOUGH THE COURSES ARE FAIRLY STEADY, THE SPEEDS
    ARE FLUCTUATING";
strout(st);
st="GREATLY DUE TO THE LOW COHERENCE LIMIT.";
strout(st);
break;
case 'd':
st=" 14:58:03        0     39.5000   147.5000        0.0
strout(st);
st=" 14:58:03     3427     39.4711   147.4707      142.0
strout(st);
st=" 15:04:23     3533     39.4682   147.4552      154.7
strout(st);
st=" 15:14:30     3449     39.4212   147.4189      153.9
strout(st);
st=" 15:38:35     3540     39.3582   147.4417      160.3
strout(st);
st=" 15:48:44     3453     39.3980   147.4815      161.6
strout(st);
st=" 15:49:31     3421     39.4021   147.4739      162.4
strout(st);
st=" 15:50:56     3479     39.4219   147.4632      160.3
strout(st);
st=" 15:52:34     3429     39.4037   147.4297      161.1
strout(st);
strtxy(4.0,12.0);
st="A VERY GOOD TRACK WITH STEADY COURSES AND SPEEDS.";
strout(st);
break;
}
}

sena() {   /* This subroutine displays the Chi-Square
              scores for various sets of data  */
char *st,*sp;
color(1);
block(0.0,0.0,100.0,100.0);
color(4);
block(0.0,15.0,100.0,90.0);
color(2);
strtxy(10.0,98.0);
strout(top);
color(5);
strtxy(5.0,86.0);
st=" RUN NUMBER 6   REGION NUMBER 6   TRACK NUMBER 9";
strout(st);
sp="  ";
strout(sp);
st="Water Time     peak      Prob      --CHI Square Scores--
```

```
    Meas Residuals";
strout(st);
st=" 23-Feb-80    Number      Score    Stagewise   Cumulative
     Deltat   Deltaf";
strout(st);
strout(sp);
switch(choice)  {
     /* Displays one of four possible sets of data  */
case 'd':
st=" 14:58:03          0         0.00       0.0000       0.0000
strout(st);
st=" 14:58:03       3427        -1.80       0.3171       0.3171
strout(st);
st=" 15:04:23       3533        -4.89       0.0162       0.1667
strout(st);
st=" 15:14:30       3449        -3.99       0.4410       0.2581
strout(st);
st=" 15:38:35       3540        -4.66       1.4096       0.2460
strout(st);
st=" 15:48:44       3453        -4.81       2.0796       0.8527
strout(st);
st=" 15:49:31       3421        -4.31       1.8945       0.8892
strout(st);
st=" 15:50:56       3479        -4.75       2.0038       0.9327
strout(st);
st=" 15:52:34       3429        -4.89       2.2893       0.9402
strout(st);
strtxy(4.0,12.0);
st="OUTSTANDING CUMULATIVE SCORE BY THE FINAL TIME PERIOD,
    THUS A VERY GOOD TRACK.";
strout(st);
break;
case 'a':
st=" 16:52:56          0         0.00       0.0000       0.0000
strout(st);
st=" 16:52:56       3542        -1.49       0.1435       0.1435
strout(st);
st=" 16:55:37       3478        -1.27       0.2159       0.1537
strout(st);
st=" 16:58:10       3548        -1.82       0.5382       0.3829
strout(st);
st=" 16:59:19       3557        -1.95       0.6547       0.4690
strout(st);
st=" 17:03:02       3461        -2.56       1.1021       0.7023
strout(st);
st=" 17:05:56       3640        -1.13       0.2371       0.6501
strout(st);
st=" 17:09:42       3582        -2.03       0.6829       0.6928
strout(st);
st=" 17:11:48       3533        -3.05       0.7928       0.7520
strout(st);
strtxy(4.0,12.0);
st="THE CUMULATIVE SCORE IS FAIR BUT NONE OF THE INDIVIDUAL
    SCORES IS REAL HIGH.";
```

```
strout(st);
break;
case 'b':
st=" 21:36:04         0         0.00        0.0000        0.0000
strout(st);
st=" 21:36:04      3560        -0.90        0.1021        0.1021
strout(st);
st=" 21:37:09      3532        -1.43        0.2341        0.1735
strout(st);
st=" 21:37:52      3571        -1.78        0.3417        0.2638
strout(st);
st=" 21:38:38      3564        -1.93        0.3176        0.4173
strout(st);
st=" 21:39:41      3551        -2.06        0.3814        0.4328
strout(st);
st=" 21:40:02      3554        -2.54        0.4027        0.4458
strout(st);
st=" 21:40:58      3584        -2.36        0.3311        0.4637
strout(st);
st=" 21:41:44      3547        -2.21        0.3517        0.4528
strout(st);
strtxy(4.0,12.0);
st="VERY LOW CUMULATIVE SCORE AND THUS A VERY POOR TRACK.";
strout(st);
break;
case 'c':
st=" 06:12:37         0         0.00        0.0000        0.0000
strout(st);
st=" 06:12:37      3325        -0.83        0.2134        0.2134
strout(st);
st=" 06:13:25      3342        -1.52        0.4327        0.3463
strout(st);
st=" 06:14:03      3394        -1.83        0.5621        0.4743
strout(st);
st=" 06:14:57      3327        -1.97        0.7631        0.6411
strout(st);
st=" 06:16:32      3319        -2.24        0.6428        0.6229
strout(st);
st=" 06:17:29      3364        -2.56        0.8817        0.7536
strout(st);
st=" 06:18:54      3318        -2.72        0.9443        0.7836
strout(st);
st=" 06:20:21      3343        -2.74        0.9218        0.7721
strout(st);
strtxy(4.0,12.0);
st="FAIR CUMULATIVE VALUES BUT LOW INDIVIDUAL VALUES DUE TO
     LOW COHERENCE LIMITS.";
strout(st);
break;
}
}

sens() {  /* This display presents surface plots */
char *st,*sp;
```

```
Q=" o o   O   o          o     ";
M="  O  oO  ";
N="     O o  ";
B="  o O ";
W="       o o      o     O      ";
E="   o   oO     o    O     o  ";
R="o            O           o  ";
T="                    o  o     ";
Y="   o                o     O  ";
color(4);
block(0.0,0.0,100.0,100.0);
color(1);
block(0.0,0.0,100.0,15.0);
block(0.0,90.0,100.0,100.0);
color(5);
block(40.0,45.0,71.5,78.0);
color(0);
strtxy(4.0,97.0);
strout(top);
 switch(choice) {
      /* Displays one of four possible surface plots */
  case 'b':
   sur1(); sur2(); sur3(); sur4();
   strtxy(4.0,12.0);
   st="JUST A LOT OF NOISE, TOO LOW COHERENCE LIMITS AND TOO
       LARGE AN AREA.";
   strout(st);
   break;
  case 'a':
   sur1(); sur3();
   strtxy(4.0,12.0);
   st="   NO PATTERN IN THIS LARGE AREA";
   strout(st);
   break;
  case 'c':
   sur1(); sur2(); sur3();
   strtxy(4.0,12.0);
   st="TOO MUCH NOISE DUE TO THE LOW COHERENCE LIMITS";
   strout(st);
   break;
  case 'd':
   sur4(); sur5();
   strtxy(4.0,12.0);
   st="GOOD SOLID PATTERN DOWN THE LEFT SIDE OF THE REGION";
   strout(st);
   break;
 }
color(4);
block(40.0,78.0,85.0,90.0);
block(71.5,45.0,100.0,85.0);
}

sur1()  {
strtxy(40.1,80.0);
```

```
strout(Q);
strout(E);
strout(Q);
strout(Y);
strout(R);
strout(E);
strout(T);
}

sur2() {
strtxy(42.0,78.5);
strout(R);
strout(T);
strout(W);
strout(E);
strout(Y);
strout(W);
strout(E);
}

sur3()  {
strtxy(41.0,81.5);
dblwid(1);
strout(Q);
strout(W);
strout(E);
strout(W);    /* Each of these short subroutines draws */
strout(R);    /* some circles on the surface plot that */
strout(Q);    /* is presently being displayed          */
strout(Y);
dblwid(0);
}

sur4()  {
strtxy(42.9,83.0);
strout(Y);
strout(R);
strout(W);
strout(W);
strout(T);
strout(Q);
}
sur5()  {
strtxy(40.0,80.0);
strout(E);
strout(E);
strout(W);
strout(R);
strout(E);
strout(Q);
strout(R);
strtxy(40.6,78.0);
dblwid(1);
strout(W);
```

```
strout(B);
strout(F);
strout(B);
strout(B);
strout(F);
strout(B);
dblwid(Ø);
strtxy(41.Ø,8Ø.9);
strout(B);
strout(M);
strout(N);
strout(M);
strout(B);
strout(N);
strout(M);
strtxy(4Ø.2,79.5);
strout(M);
strout(N);
strout(M);
strout(M);
strout(B);
strout(N);
strout(M);
}

changcolor()  {
      /* Establishes a new color table  */
 int a[16];
 a[Ø] = triple(Ø,Ø,Ø);
 a[1] = triple(15,Ø,Ø);
 a[2] = triple(Ø,15,Ø);
 a[3] = triple(Ø,Ø,15);
 a[4] = triple(15,5,Ø);
 a[5] = triple(2,Ø,6);
 a[6] = triple(Ø,1Ø,Ø);
 a[7] = triple(2,6,1Ø);
 a[8] = triple(6,1Ø,2);
 a[9] = triple(1Ø,2,6);
 a[1Ø] = triple (13,5,2);
 a[11] = triple(5,2,13);
 a[12] = triple (2,13,5);
 a[13] = triple(15,Ø,15);
 a[14] = triple(1Ø,4,4);
 a[15] = triple(15,15,15);
 clrtbl(1Ø,a);
 return;
}   /* End of the Senario   */
```

BIBLIOGRAPHY

Alter, S. Why is man-computer interaction important for decision support systems? Interfaces, February 1977, 7(2), 109-115.

Baker, J.D. Quantitative modeling of human performance in information systems. Ergonomics, 1970, 13, 645-664.

Baldwin, J.T. & Siklossy, L. An unobtrusive computer monitor for multi-step problem solving. International Journal of Man-Machine Studies, 1977, 9, 349-362.

Bois, S.J. User behavior on an interactive computer system. IBM Systems Journal, 1974, 13, 2-18.

Caruso, D.E. Tutorial programs for operation of on-line retrieval systems. Journal of Chemical Documentation, 1970, 10, 98-105.

Chapanis, A. On the allocation of functions between men and machines. Occupational Psychology, 1965, 39,1-11. (NTIS No. AD 626311).

Christ, R.E. Review and analysis of color coding research for visual displays. Human Factors, 1975, 17, 542-570.

Dorris, A.L., Sadosky, T.L.,& Connolly, T. Varying data and information in a decision making task. Ergonomics, 1977, 20, 643-649.

Goodwin, N.C. Cursor positioning on an electronic display using lightpen, lightgun, or keyboard for three basic tasks. Human Factors, 1975, 17, 289-295.

Gould, J.D. Visual factors in the design of computer-controlled CRT displays. Human Factors, 1968, 10, 359-375.

Meister, D. & Farr, D.E. The utilization of human factors information by designers. Human Factors, 1967, 9, 71-87.

Morrill, C.S. Computer-aided instruction as part of a management information system. Human Factors, 1967, 9, 251-256.

Morrill, C.S., Goodwin, N.C.,& Smith,S.L. User input mode and computer-aided instruction. Human Factors, 1968, 10, 225-232.

Newsted, P.R. & Wynne, B.E. Augmenting man's judgment with interactive computer systems. International Journal of Man-Machine Studies, 1976, 8, 29-59.

Schutz, H.G. An evaluation of formats for graphic trend displays: Experiment II. Human Factors, 1961, 3, 99-107.

Siegal, A.I. & Fischl, M.A. Dimensions of visual information displays. Journal of Applied Psychology, 1971, 55, 470-476.

Smith, S.L. Color coding and visual separability in information displays. Journal of Applied Psychology, 1963, 47, 358-364.

Smith, S.L. Color-coded displays for data-processing systems. Electro-Technology, April 1963, 71(4), 63-69.

Vitz, P.C. Preference for different amounts of visual complexity. Behavioral Science, 1966, 11, 105-114.

Waksman, A. The interface problem in interactive systems. Behavior Research Methods and Instrumentation, 1794, 6, 235-246.

INITIAL DISTRIBUTION LIST

		No. copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Ivan Hall PME 124 NAVELEX 6W67 NC1 Washington D.C. 20360	1
3.	Carl Andriani PME 124 NAVELEX 2W84 NC1 Washington D.C. 20360	1
4.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
5.	Department Chairman, Code 52BZ Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
6.	Professor G. A. Rahe, Code 52RA Department of Computer Science Naval Postgraduate School Monterey, California 93940	2
7.	Professor B. J. MacLennan, Code 52ML Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
8.	Professor William L. Flynn Milwaukee School of Engineering 1025 N. Milwaukee Avenue Milwaukee, Wisconsin 53001	1
9.	Lt. Patrick W. Flynn, USN 86 Norseman Drive Portsmouth, Rhode Island 02871	2

Thesis
F5275 Flynn 187068
c.1
An embedded tutorial for a tactical system.